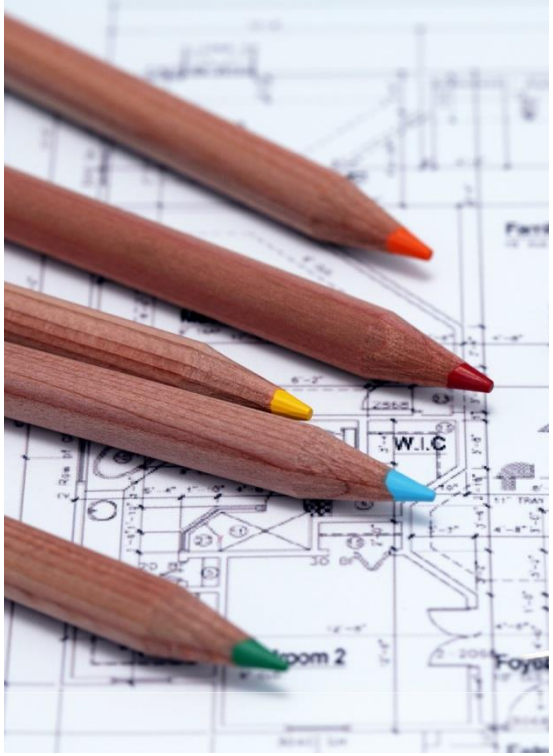


ARS

Java Server Faces

Internationalisierung

Überblick

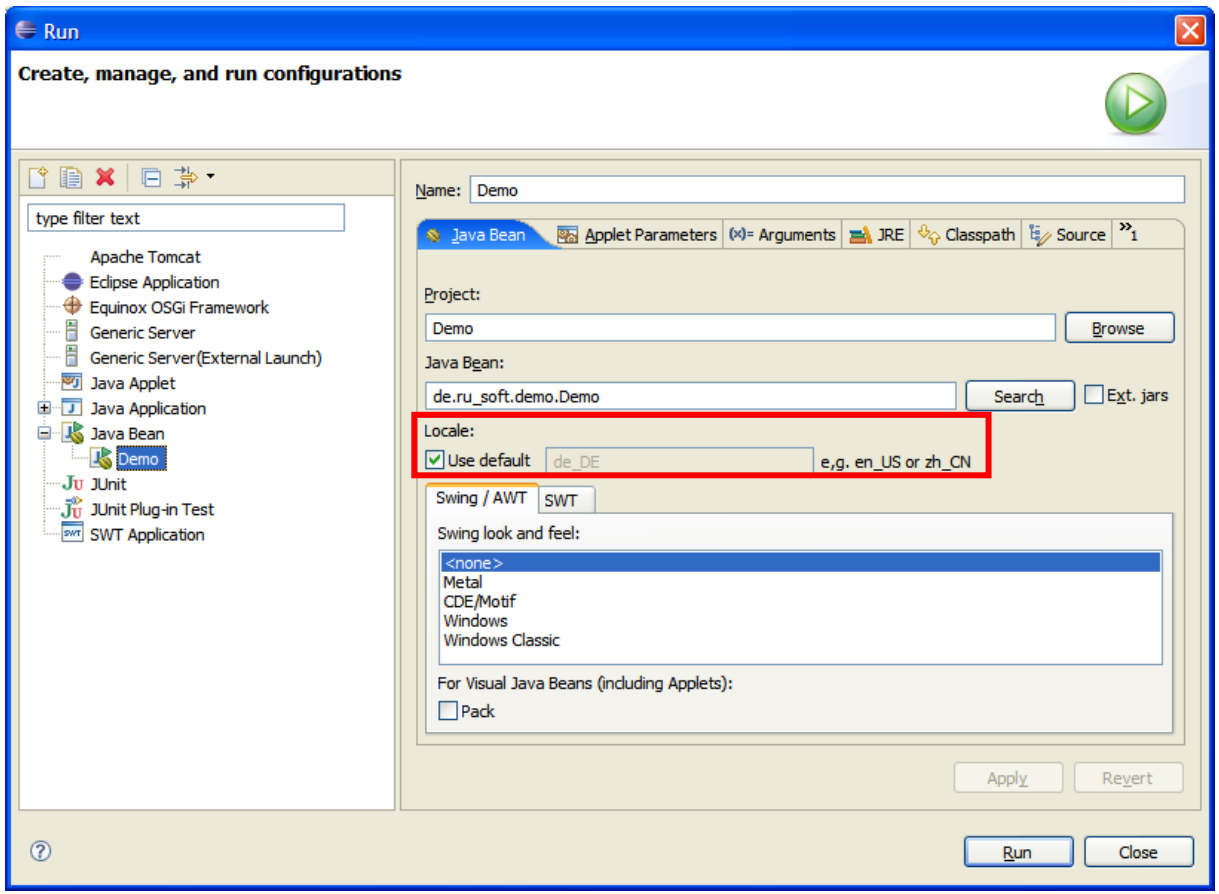
- Grundlagen
  - Locales
  - Internationale Formatierung
  - Sprachdateien
  - Parameter für Zeichenketten
- Internationalisierung mit JSF
  - Internationalisierung im Web
  - Meldungen für Standardkomponenten
- Internationalisierung mit Bean Validation (JSR-303)

i18n

- Länderspezifische Aus- und Eingaben werden zentral verwaltet
  - Sprache
  - Zahlenformatierungen
  - Datumsformatierungen
  - Zeichensatz
- Landesinformationen: `java.util.Locale`
  - Sprache (`language`), ISO 639, Kleinbuchstaben
  - Land (`country`), ISO 3166, Großbuchstaben, optional
  - Plattform (`variant`) , Großbuchstaben, optional
  - Z.B. en_US, de_DE, zh_TW
  - Aktuelles Locale: `Locale.`*`getDefault()`*

Setzen einer Sprache

- Standard ist Einstellung des Betriebssystems
- `java -Duser.language=en_US MyApp`
- Launch Configuration (Eclipse)



Formatierung von Zahlen

- Standardformatierung der Locale

```
Locale currentLocale = Locale.getDefault();
NumberFormat numberFormatter
    = NumberFormat.getNumberInstance(currentLocale);

Double amount = new Double(345987.246);
String amountOut = numberFormatter.format(amount);
System.out.println(amountOut);
```

- **Ausgabe**

```
345 987,246      (fr_FR)
345.987,246      (de_DE)
345,987.246      (en_US)
```

Texteigenschaften

- Schlecht, da beispielsweise keine Umlaute berücksichtigt werden
  - `if ((ch>='a' && ch<='z') || (ch>='A' && ch<='Z'))`
- Besser
  - `if (Character.isLetter(ch))`
- Weitere Methoden von `java.lang.Character`
  - `isDigit, isLetter, isLetterOrDigit`
  - `isLowerCase, isUpperCase`
  - `isSpaceChar`
  - `isDefined`
- Sortierung von Strings

```
Collator col = Collator.getInstance(locale);
if (col.compare("Hut","Hüte") >0) {
  System.out.println("Hut vor Hüte");
}
```

Worttrennung und Texterkennung

- Trennung zwischen Wörtern und innerhalb von Wörtern
- Erkennung von logischen Textgrenzen
  - Zeichen
  - Wort
  - Satz
  - Zeile
- Beispiel: (Ausgabe: 0 3 4 7 8 12)

```
BreakIterator it = BreakIterator.getWordInstance(Locale.getDefault());
it.setText("Hut vor Hüte");
for(int boundary=it.first(); boundary!=BreakIterator.DONE;
boundary = it.next()) {
   System.out.println (boundary);
}
```

Mehrsprachigkeit

- Konzept:
  - Zeichenketten werden zentral verwaltet
  - Jede Zeichenkette bekommt einen Namen (key)
  - Pro Sprache eine Sprachdatei
  - Sprachdateien enthalten dieselben Schlüssel (keys)
  - Oberflächenkomponenten greifen auf die Schlüssel einer Sprachdatei zu, um an die Zeichenketten zu gelangen

ARS

Grundlagen

`java.util.ResourceBundle`

- Sammlung von sprachabhängigen Informationen
- Quellen der Informationen
  - Externe Datei (`PropertyResourceBundle`)
  - Java-Klassen (`ListResourceBundle`)
  - Eigene ResourceBundle-Ableitungen

ARS

Grundlagen

`java.util.PropertyResourceBundle`

- Sammlung von Zeichenketten der Oberfläche
- Einbinden externer Properties-Dateien
- Erzeugung über Factory Pattern

```
ResourceBundle bundle =
                    ResourceBundle.getBundle("labels");
```

Datei labels.properties	Datei labels_en.properties
`button.text.cancel=Abbrechen`	`button.text.cancel=Cancel`
`button.text.finish=Fertigstellen`	`button.text.finish=Finish`

Abfragen von Werten

- ResourceBundle bietet Methoden zum Lesen
  - `getString(String key)`
  - `getObject(String key)` für Java-Objekte
  - `getStringArray(String key)` für Java-Objekte

```
btnCancel.setText(
   bundle.getString("button.text.cancel")
);
```

```
// {"page", new DecimalFormat("#,##0")}
NumberFormat nf = (NumberFormat)rb.getObject("page");
txtPageNumber.setText(nf.format(pageNumber));
```

Parameter für Werte

- Zeichenketten können Parameter enthalten
  - Nummerierung `{0}, {1}, ...`
  - Angabe von Formaten `{0,date,long}, {1,number,currency}, ...`
- Setzen der Parameter mit `java.text.MessageFormat`

```
// datum=Heute haben wir {0}.
MessageFormat formatter =
   new MessageFormat(bundle.getString("datum"));
formatter.setLocale(Locale.getDefault());
String text = formatter.format(new Object[]{
   new Date()
});
```

Parameter für Werte (komplexeres Beispiel)

- https://wiki.imise.uni-leipzig.de/Themen/Java/I18N

```
planet=Erde
template=Am {0,date,long} um {0,time,long} {1}
         vom Planeten {2} für {3,number,currency}
         verkauft.
noShips=wurden keine Raumschiffe
oneShip=wurde ein Raumschiff
multipleShips=wurden {1,number,integer} Raumschiffe
```

Parameter für Werte (komplexeres Beispiel)

```
MessageFormat formatter = new MessageFormat(
   bundle.getString("template")
);
formatter.setLocale(Locale.getDefault());
// {1} wird bedingt formatiert
formatter.setFormatByArgumentIndex(1, new ChoiceFormat(
   new double[]{0, 1, 2}, // Intervallgrenzen
   new String[] {
      bundle.getString("noShips"),
      bundle.getString("oneShip"),
      bundle.getString("multipleShips")
   }
));
```

Parameter für Werte (komplexeres Beispiel)

```
String text = formatter.format(new Object[]{
    /*{0}*/ new Date(),
    /*{1}*/ 5,
    /*{2}*/ bundle.getString("planet"),
    /*{3}*/ 1000
}));
```

→ Am 7. März 2007 um 19:23:01 CET wurden 5 Raumschiffe vom Planeten Erde für 1.000,00 € verkauft.

Fallstricke bei Internationalisierung

- Grammatische Unterschiede in der Satzstellung (Stringkonkatenation)
- Blindes Internationalisieren aller Strings
- Arbeiten mit festen Textfeldlängen
  - Postleitzahlen variieren in verschiedenen Ländern

Internationalisierung mit JSF

Allgemeines

```
GET /training/index.html HTTP/1.0
Accept-Language: en-us
…
```

- HTTP
  - Bevorzugte Sprachen des Benutzers per Request-Header
  - Einstellung im Browser
- JSF
  - Konfiguration unterstützter Sprachen in `faces-config.xml`
  - Konfiguration von Resource Bundles in `faces-config.xml`
    - 1 Message Bundle
      - »Keys in FacesMessage: Validierung, Konvertierung, …
    - Mehrere Resource Bundles mit Variablennamen
      - »Zugriff in Facelet durch Expression
  - Auswahl der Sprache pro Request durch FacesServlet

Internationalisierung mit JSF

Konfiguration der Sprachen

- Datei `faces-config.xml`

```
<application>
  <resource-bundle>
    <base-name>de.ars.training.modules.messages</base-name>
    <var>modulesbundle</var>
  </resource-bundle>
  <message-bundle>de.ars.training.messages</message-bundle>
  <locale-config>
    <default-locale>de</default-locale>
    <supported-locale>en</supported-locale>
  </locale-config>
</application>
```

Internationalisierung mit JSF

Zugriff auf ResourceBundles

- Facelet
  - Kein Zugriff auf MessageBundle
  - Zugriff auf Application Resource Bundles nach Variablennamen

```
<h:outputFormat value="#{modulesbundle.text}">
  <f:param value="#{user.name}">
</h:outputFormat>
```

- Programmatisch

```
FacesContext ctx = FacesContext.getCurrentInstance();
Application app = ctx.getApplication();
String title = app.getResourceBundle(ctx,"modulesbundle").getString("title");
String message = ResourceBundle.getBundle(
                    app.getMessageBundle(),
                    ctx.getViewRoot().getLocale(),
                    app.getClass().getClassLoader()
                  ).getString("myMessage");
```

Ermittlung des Locales

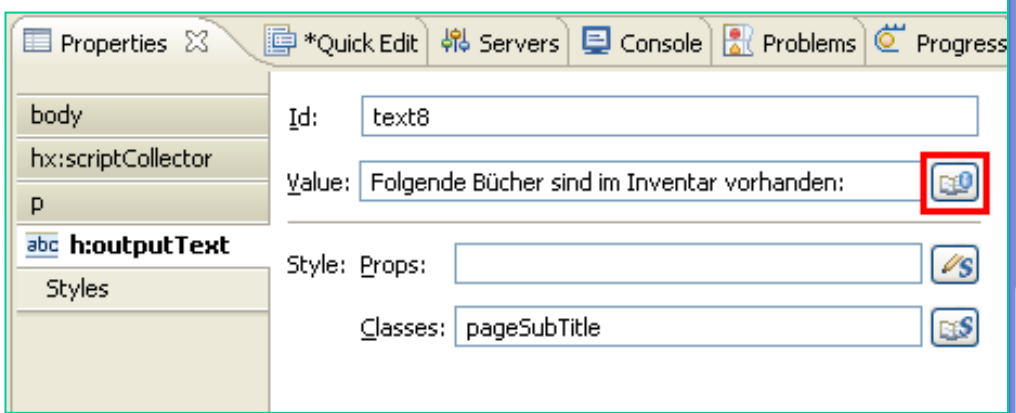
- FacesServlet auf Basis von
  - Konfiguration in `faces-config.xml`
  - Request-Header mit Spracheinstellungen am Browser
- Zugriff über

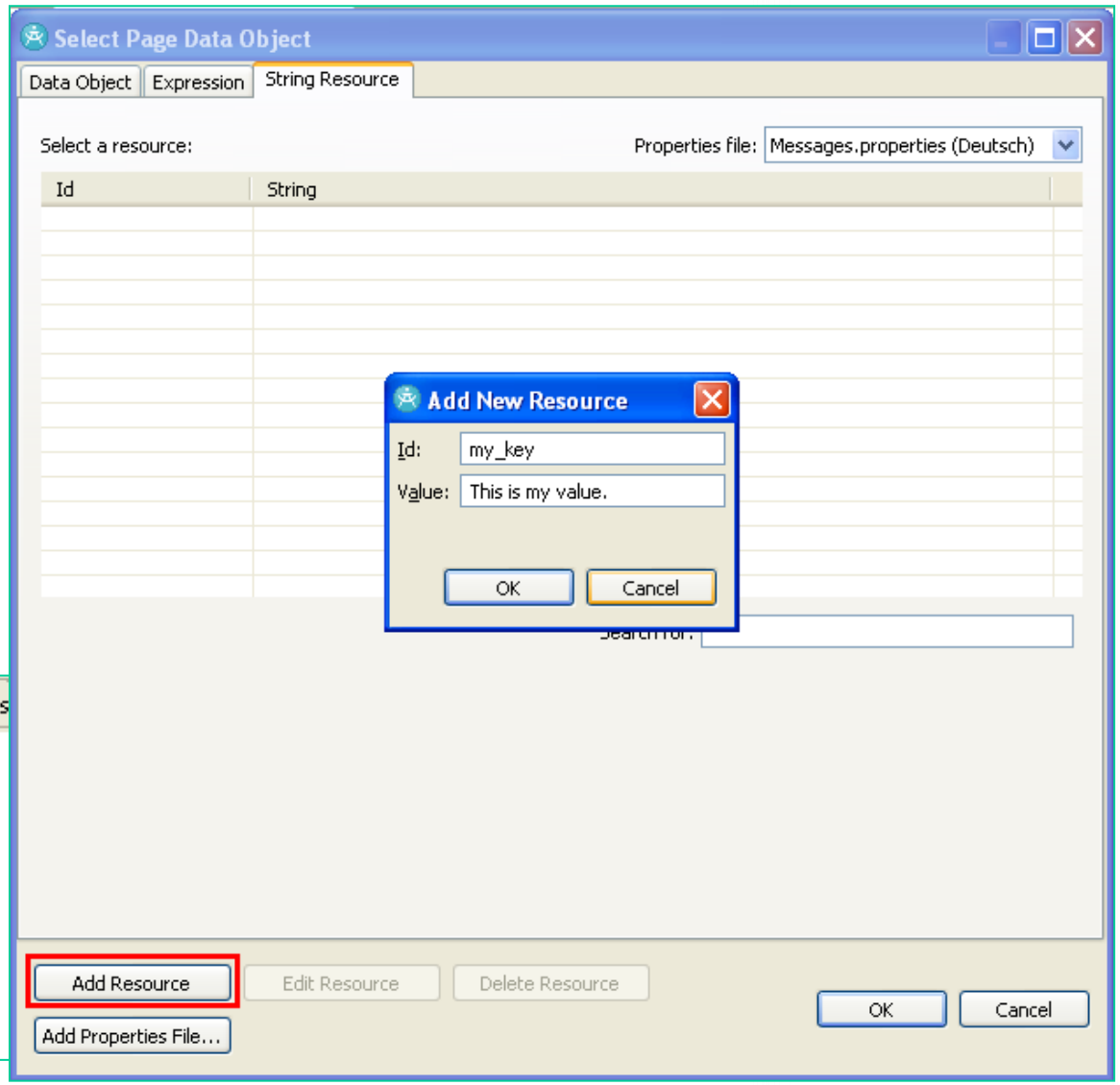
```
FacesContext.getCurrentInstance().getViewRoot().getLocale();
```

Internationalisierung mit JSF

Tooling im RAD/RSA

- Dialog zum Setzen einer Beschriftung (Properties-View)
  - Auswahl oder Erstellen einer Properties-Datei
  - Auswahl oder Erstellen einer Resource (Key/Value)





ARS

Internationalisierung mit Bean Validation (JSR-303)

- Mechanismus unabhängig von JSF
- Angabe von Keys über Annotationen

```
public class Person {

  @Size(min=2, message="{person.firstName.size}")
  private String lastName;

}
```

- Ablegen der Zeichenketten in `ValidationMessages.properties` bzw. `ValidationMessages_<Locale>.properties`
  - Platzhalter je nach Annotation

```
person.firstName.size = Please specify a minimum of {min} characters.
```

Kontrollfragen

- Wie werden Oberflächen in Java prinzipiell mehrsprachig gestaltet?
- Wo wird die Sprache bei Desktop- bzw. Webapplikationen eingestellt?
- Welche Einstellungsmöglichkeiten bzgl. Mehrsprachigkeit bietet JSF?