

ENTWICKLUNG EINER FORMULARANWENDUNG MIT KOMPATIBILITÄTSVALIDIERUNG DER EINFACH- UND MEHRFACHAUSWAHL-EINGABEFELDER

Vorgelegt von:

Alexander Johr

Meine Adresse

Erstprüfer: Prof. Jürgen Singer Ph.D.
Zweitprüfer: Prof. Daniel Ackermann
Datum: 02.11.2020

Teil I

Implementierung

0.1 Schritt 3

```
19 final onSelect<ChoiceType> onSelect;
20 final OnDeselect<ChoiceType> onDeselect;
21 final String? errorText;
22
23 SelectionCard(
24   {required this.title,
25     required Iterable<ChoiceType> initialValue,
26     required this.allChoices,
27     required this.onSelect,
28     required this.onDeselect,
29     this.errorText,
30     Key? key})
```

Listing 1: errorText wird der SelectionCard hinzugefügt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/widgets/selection_card.dart](#)

```
63   onTap: navigateToSelectionScreen,
64 ),
65 if (errorText != null)
66   Padding(
67     padding: const EdgeInsets.all(8.0),
68     child: Text(errorText!,
69       style:
70         const TextStyle(fontSize: 12.0, color: Colors.red)),
71   )
```

Listing 2: errorText wird ausgegeben, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/widgets/selection_card.dart](#)

```
215   mini: true,
216   heroTag: 'save_draft_floating_action_button',
217   child: const Icon(Icons.paste, color: Colors.white),
218   backgroundColor: Colors.orange,
219   onPressed: saveDraft,
220 ),
221 const SizedBox(
222   height: 10,
223 ),
224 FloatingActionButton(
225   tooltip: saveMassnahmeTooltip,
226   heroTag: 'save_floating_action_button',
227   child: const Icon(Icons.check, color: Colors.white),
228   onPressed: validateAndSave,
229 )
230 ],
```

Listing 3: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

25     ..showSnackBar(
26         const SnackBar(content: Text('Entwurf wird gespeichert ...')));
27
28     var draft = vm.model.rebuild((b) =>
29         b.letzteBearbeitung.letzterStatus = LetzterStatus.bearb.abbreviation);
30
31     model.putMassnahmeIfAbsent(draft);
32     Navigator.of(context).pop();
33 }
34
35 void saveRecordAndGoBackToOverviewScreen() {
36     ScaffoldMessenger.of(context)

```

Listing 4: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

20 final formKey = GlobalKey<FormState>();
21
22 void saveDraft() {
23     ScaffoldMessenger.of(context)

```

Listing 5: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

48     children: [
49         Text(
50             'Fehler im Formular trotz Status "${LetzterStatus.fertig.description}"',
51             const SizedBox(width: 4),
52             ElevatedButton(
53                 onPressed: saveDraft,
54                 child: Padding(
55                     padding: const EdgeInsets.fromLTRB(4, 4, 8, 4),
56                     child: Row(
57                         children: const [
58                             Icon(Icons.paste, color: Colors.white),
59                             SizedBox(width: 4),
60                             Text(
61                                 "Entwurf speichern?",
62                                 style: TextStyle(fontSize: 18.0, color: Colors.white),
63                             ),
64                         ],
65                     ),
66                 ),
67             ),
68         ],
69     ));
70 }
71
72 bool inputsAreValidOrNotMarkedFinal() {
73     if (vm.letzterStatus.value != LetzterStatus.fertig) {

```

Listing 6: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

75 }
76
77 if (formKey.currentState!.validate()) {
78   return true;
79 }
80
81 return false;
82 }
83
84 Future<bool> validateAndSave() {
85   if (inputsAreValidOrNotMarkedFinal()) {

```

Listing 7: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

87   return Future.value(true);
88 } else {
89   showValidationError();
90   return Future.value(false);
91 }
92 }
93
94 Widget createMassnahmenTitelTextFormField() {
95   final focusNode = FocusNode();

```

Listing 8: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

```

132 return FormField(
133   validator: (_) {
134     if (selectionViewModel.value == null) {
135       return "Feld ${allChoices.name} enthält keinen Wert!";
136     }
137
138     Iterable<Choice> choices = {
139       if (selectionViewModel.value != null) selectionViewModel.value!
140     };
141
142     if (choices.isEmpty) {
143       return "Feld ${allChoices.name} enthält keinen Wert!";
144     }
145
146     return null;
147   },
148   builder: (field) => SelectionCard<ChoiceType>(
149     title: allChoices.name,
150     allChoices: allChoices,
151     initialValue: {
152       if (selectionViewModel.value != null)
153         selectionViewModel.value!
154     },
155     onSelect: (selectedChoice) =>
156       selectionViewModel.value = selectedChoice,
157     onDeselect: (selectedChoice) => selectionViewModel.value = null,
158     errorText: field.errorText,
159   ));
160 }
161
162 return Scaffold(
163   appBar: AppBar(

```

Listing 9: Die Maßnahmencharakteristika Selektionskarten werden ergänzt, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-3/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart](#)

Teil II

Anhang

A Schritt 2 Anhang

```

5 class FoerderklasseChoice extends Choice {
6   static final oelb = FoerderklasseChoice("oelb", "Ökolandbau");
7   static final azl = FoerderklasseChoice("azl", "Ausgleichszulage");
8   static final ea = FoerderklasseChoice("ea", "Erschwernisausgleich");
9   static final aukm_nur_vns = FoerderklasseChoice("aukm_nur_vns",
10     "Agrarumwelt-(und Klima)Maßnahme: nur Vertragsnaturschutz");
11   static final aukm_ohne_vns = FoerderklasseChoice("aukm_ohne_vns",
12     "Agrarumwelt-(und Klima)Maßnahmen, tw. auch mit Tierwohlaspekten, aber OHNE
13     ↳ Vertragsnaturschutz");
14   static final twm_ziel = FoerderklasseChoice(
15     "twm_ziel", "Tierschutz/Tierwohlmaßnahmen mit diesem als Hauptziel");
16   static final contact =
17     FoerderklasseChoice("contact", "bitte um Unterstützung");
18   FoerderklasseChoice(String abbreviation, String description,
19     {bool Function(Set<Choice> choices)? condition})
20     : super(abbreviation, description);
21 }
22
23 final foerderklasseChoices = Choices<FoerderklasseChoice>({
24   FoerderklasseChoice.oelb,
25   FoerderklasseChoice.azl,
26   FoerderklasseChoice.ea,
27   FoerderklasseChoice.aukm_nur_vns,
28   FoerderklasseChoice.aukm_ohne_vns,
29   FoerderklasseChoice.twm_ziel,
30   FoerderklasseChoice.contact
31 }, name: "Förderklasse");
32
33 class KategorieChoice extends Choice {
34   static final zf_us =
35     KategorieChoice("zf_us", "Anbau Zwischenfrucht/Untersaat");
36   static final anlage_pflege =
37     KategorieChoice("anlage_pflege", "Anlage/Pflege Struktur");
38   static final dungmang = KategorieChoice("dungmang", "Düngemanagement");
39   static final extens = KategorieChoice("extens", "Extensivierung");
40   static final flst = KategorieChoice("flst", "Flächenstilllegung/Brache");
41   static final umwandlg = KategorieChoice("umwandlg", "Nutzungsumwandlung");
42   static final bes_kult_rass = KategorieChoice(
43     "bes_kult_rass", "Förderung bestimmter Rassen / Sorten / Kulturen");
44   static final contact = KategorieChoice("contact", "bitte um Unterstützung");
45
46   KategorieChoice(String abbreviation, String description)
47     : super(abbreviation, description);
48 }
49
50 final kategorieChoices = Choices<KategorieChoice>({
51   KategorieChoice.zf_us,
52   KategorieChoice.anlage_pflege,
53   KategorieChoice.dungmang,
54   KategorieChoice.extens,
55   KategorieChoice.flst,
56   KategorieChoice.umwandlg,
57   KategorieChoice.bes_kult_rass,
58   KategorieChoice.contact
59 }, name: "Kategorie");

```

Listing 10: Die Mengen foerderklasseChoices und kategorieChoices , Quelle: Eigenes Listing, Datei: Quellcode/Schritt-2/conditional_form/lib/choices/choices.dart

```

61 class ZielflaecheChoice extends Choice {
62   static final ka = ZielflaecheChoice("ka", "keine Angabe/Vorgabe");
63   static final al = ZielflaecheChoice("al", "AL");
64   static final gl = ZielflaecheChoice("gl", "GL");
65   static final lf = ZielflaecheChoice("lf", "LF");
66   static final dk_sk = ZielflaecheChoice("dk_sk", "DK/SK");
67   static final hff = ZielflaecheChoice("hff", "HFF");
68   static final biotop_le =
69     ZielflaecheChoice("biotop_le", "Landschaftselement/Biotop o.Ä.");
70   static final wald = ZielflaecheChoice("wald", "Wald/Forst");
71   static final contact = ZielflaecheChoice("contact", "bitte um Unterstützung");
72
73   ZielflaecheChoice(String abbreviation, String description)
74     : super(abbreviation, description);
75 }
76
77 final zielflaecheChoices = Choices<ZielflaecheChoice>({
78   ZielflaecheChoice.ka,
79   ZielflaecheChoice.al,
80   ZielflaecheChoice.gl,
81   ZielflaecheChoice.lf,
82   ZielflaecheChoice.dk_sk,
83   ZielflaecheChoice.hff,
84   ZielflaecheChoice.biotop_le,
85   ZielflaecheChoice.wald,
86   ZielflaecheChoice.contact
87 }, name: "Zielfläche");
88
89 class ZieleinheitChoice extends Choice {
90   static final ka = ZieleinheitChoice("ka", "keine Angabe/Vorgabe");
91   static final m3 = ZieleinheitChoice("m3", "m3 (z.B. Gülle)");
92   static final pieces =
93     ZieleinheitChoice("pieces", "Kopf/Stück (z.B. Tiere oder Bäume)");
94   static final gve = ZieleinheitChoice("gve", "GV/GVE");
95   static final rgve = ZieleinheitChoice("rgve", "RGV");
96   static final ha = ZieleinheitChoice("ha", "ha");
97   static final contact = ZieleinheitChoice("contact", "bitte um Unterstützung");
98
99   ZieleinheitChoice(String abbreviation, String description)
100     : super(abbreviation, description);
101 }
102
103 final zieleinheitChoices = Choices<ZieleinheitChoice>({
104   ZieleinheitChoice.ka,
105   ZieleinheitChoice.m3,
106   ZieleinheitChoice.pieces,
107   ZieleinheitChoice.gve,
108   ZieleinheitChoice.rgve,
109   ZieleinheitChoice.ha,
110   ZieleinheitChoice.contact
111 }, name: "Zieleinheit");

```

Listing 11: Die Mengen zielflaecheChoices und zieleinheitChoices, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-2/conditional_form/lib/choices/choices.dart](#)

```

113 class ZielsetzungLandChoice extends Choice {
114     static final ka = ZielsetzungLandChoice("ka", "keine Angabe/Vorgabe");
115     static final bsch = ZielsetzungLandChoice("bsch", "Bodenschutz");
116     static final wsch = ZielsetzungLandChoice("wsch", "Gewässerschutz");
117     static final asch = ZielsetzungLandChoice("asch", "Spezieller Artenschutz");
118     static final biodiv = ZielsetzungLandChoice("biodiv", "Biodiversität");
119     static final struktviel =
120         ZielsetzungLandChoice("struktviel", "Erhöhung der Strukturvielfalt");
121     static final genet_res = ZielsetzungLandChoice("genet_res",
122         "Erhaltung genetischer Ressourcen (Pflanzen, z. B. im Grünland, und Tiere, z. B.
        ↳ bedrohte Rassen)");
123     static final tsch = ZielsetzungLandChoice(
124         "tsch", "Tierschutz/Maßnahmen zum Tierwohl im Betrieb");
125     static final klima = ZielsetzungLandChoice("klima", "Klima");
126     static final contact =
127         ZielsetzungLandChoice("contact", "bitte um Unterstützung");
128
129     ZielsetzungLandChoice(String abbreviation, String description)
130         : super(abbreviation, description);
131 }
132
133 final _zielsetzungLandChoices = {
134     ZielsetzungLandChoice.ka,
135     ZielsetzungLandChoice.bsch,
136     ZielsetzungLandChoice.wsch,
137     ZielsetzungLandChoice.asch,
138     ZielsetzungLandChoice.biodiv,
139     ZielsetzungLandChoice.struktviel,
140     ZielsetzungLandChoice.genet_res,
141     ZielsetzungLandChoice.tsch,
142     ZielsetzungLandChoice.klima,
143     ZielsetzungLandChoice.contact
144 };
145
146 final hauptzielsetzungLandChoices = Choices<ZielsetzungLandChoice>(
147     _zielsetzungLandChoices,
148     name: "Hauptzielsetzung Land");

```

Listing 12: Die Menge hauptzielsetzungLandChoices, Quelle: Eigenes Listing, Datei: [Quellcode/Schritt-2/conditional_form/lib/choices/choices.dart](#)