


# Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

 base repository: OpenOdonto/sco

base: master

...

head repository: Felipe-Gs/sco

compare: Long-Method---#-1

✓ Able to merge.

These branches can be automatically merged.


2 commits

1 file changed


1 contributor

Commits on Jun 14, 2023

Refactor QuestaoQuestionarioConverter

 Felipe-Gs committed on Jun 14

refactor: Magical number removido

 Felipe-Gs committed on Jun 14

Showing 1 changed file with 55 additions and 51 deletions.

Split

Unified

106

...

...nto-web/src/main/java/br/ueg/openodonto/visao/converter/QuestaoQuestionarioConverter.java

15

15

import br.ueg.openodonto.controle.ManterQuestionarioAnamnese;

16

16

import br.ueg.openodonto.util.beam.QuestionarioQuestaoAdapter;

17

17

18

- public class QuestaoQuestionarioConverter implements Converter{

19

-

20

-

21

- private Map<String,Integer> indexMap;

22

- private boolean isIndexed;

23

-

24

- private void doIndexMap(FacesContext context, UIComponent component){

25

- indexMap = new HashMap<String, Integer>();

26

- UIOrderingList orderingList = (UIOrderingList) component;

27

- String clientId = orderingList.getBaseClientId(context);

28

- ExternalContext externalContext = context.getExternalContext();

29

- Map<String, String[]> requestParameterValuesMap =

externalContext.getRequestParameterValuesMap();

```

30         String[] strings = (String[])
requestParameterValuesMap.get(clientId);
31         if (strings != null && strings.length != 0) {
32             for(String string : strings){
33                 boolean fw1 =
Character.isLetter(string.charAt(0));
34                 boolean sw1 =
Character.isLetter(string.charAt(1));
35                 int indexDivisor = string.indexOf(":");
36                 int indexReal = fw1 ? sw1 ? 2 : 1 : 0;
37                 int indexVirtual = indexDivisor+1;
38                 Integer real =
Integer.valueOf(string.substring(indexReal,indexDivisor));
39                 String virtual = string.substring(indexVirtual);
40                 indexMap.put(virtual, real);
41             }
42         }
43         isIndexed = true;
44     }
45
46     @Override
47     public Object getAsObject(FacesContext context, UIComponent component,
String value) {
48         if(!isIndexed){
49             doIndexMap(context,component);
50         }
51         Integer id = indexMap.get(value);
52         ManterQuestionarioAnamnese mBean = getMbean(context);
53         List<QuestionarioQuestaoAdapter> questoes =
mBean.getManageQuestao().getQuestoesAdapter();
54         QuestionarioQuestaoAdapter questao = questoes.get(id);
55         return questao;
56     }
57
58     @Override
59     public String getAsString(FacesContext context, UIComponent component,
Object value) {
60         QuestionarioQuestaoAdapter adapter =
(QuestionarioQuestaoAdapter) value;
61         return adapter.getQqa().getIndex().toString();
62     }
63
64     private ManterQuestionarioAnamnese getMbean(FacesContext context){
65         ELContext elctx = context.getELContext();
66         ManterQuestionarioAnamnese mBean =
(ManterQuestionarioAnamnese)elctx.getELResolver().getValue(elctx, null,
"manterQuestionarioAnamnese");
67         return mBean;
68     }
18 + public class QuestaoQuestionarioConverter implements Converter {
19 +
20 +     private Map<String, Integer> indexMap;

```

69

```

21 +     private boolean isIndexed;
22
23 +     private static final int INDEX_REAL_FIRST_LETTER = 1;
24 +     private static final int INDEX_REAL_SECOND_LETTER = 2;
25 +     private static final int INDEX_REAL_DEFAULT = 0;
26 +
27 +     private Map<String, Integer> createIndexMap(UIOrderingList orderingList,
28 + Map<String, String[]> requestParameterValuesMap, String clientId) {
29 +
30 +         String[] strings = requestParameterValuesMap.get(clientId);
31 +         if (strings != null && strings.length != 0) {
32 +             for (String string : strings) {
33 +                 boolean firstCharIsLetter =
34 +                     Character.isLetter(string.charAt(0));
35 +                 boolean secondCharIsLetter =
36 +                     Character.isLetter(string.charAt(1));
37 +                 int indexDivisor = string.indexOf(":");
38 +                 int indexReal = firstCharIsLetter ? (secondCharIsLetter ?
39 +                     INDEX_REAL_SECOND_LETTER : INDEX_REAL_FIRST_LETTER) : INDEX_REAL_DEFAULT;
40 +                 int indexVirtual = indexDivisor + 1;
41 +                 Integer real = Integer.valueOf(string.substring(indexReal,
42 +                     indexDivisor));
43 +                 String virtual = string.substring(indexVirtual);
44 +                 indexMap.put(virtual, real);
45 +             }
46 +         }
47 +         return indexMap;
48 +     }
49 +
50 +     private void doIndexMap(FacesContext context, UIComponent component) {
51 +         indexMap = createIndexMap(context, component);
52 +         isIndexed = true;
53 +     }
54 +
55 +     @Override
56 +     public Object getAsObject(FacesContext context, UIComponent component,
57 + String value) {
58 +         if (!isIndexed) {
59 +             doIndexMap(context, component);
60 +         }
61 +         Integer id = indexMap.get(value);
62 +         ManterQuestionarioAnamnese mBean = getMbean(context);
63 +         List<QuestionarioQuestaoAdapter> questoes =
64 +             mBean.getManageQuestao().getQuestoesAdapter();
65 +         QuestionarioQuestaoAdapter questao = questoes.get(id);
66 +         return questao;
67 +     }
68 +
69 +     @Override
70 +     public String getAsString(FacesContext context, UIComponent component,
71 + Object value) {

```

```
65 +     QuestionarioQuestaoAdapter adapter = (QuestionarioQuestaoAdapter) value;
66 +     return adapter.getQqa().getIndex().toString();
67 + }
68 +
69 +     private ManterQuestionarioAnamnese getMbean(FacesContext context) {
70 +         ELContext elctx = context.getELContext();
71 +         ManterQuestionarioAnamnese mBean = (ManterQuestionarioAnamnese)
72 +             elctx.getELResolver().getValue(elctx, null, "manterQuestionarioAnamnese");
73 +         return mBean;
74 +     }
70 74 }
```