

## ReactDomainmodelGenerator.xtend

```
2 * generated by Xtend 2.10.0
4 package de.hsh.generator
5
6 import org.eclipse.emf.ecore.resource.Resource
7 import org.eclipse.xtext.generator.AbstractGenerator
8 import org.eclipse.xtext.generator.IFileSystemAccess2
9 import org.eclipse.xtext.generator.IGeneratorContext
10 import de.hsh.reactDomainmodel.ReactDomainModel
11 import de.hsh.reactDomainmodel.Paper
12 import de.hsh.reactDomainmodel.ReactMaterialUI
13
14 /**
15  * Generates code from your model files on save.
16  *
17  * See https://www.eclipse.org/Xtext/documentation/303\_runtime\_concepts.html#code-generation
18  */
19 class ReactDomainmodelGenerator extends AbstractGenerator {
20
21
22
23
24     override void doGenerate(Resource resource,
25         IFileSystemAccess2 fsa, IGeneratorContext context) {
26         doImports()
27         for(e:
28             resource.allContents.toIterable().filter(ReactDomainModel)){
29                 checkReactModelObject(e)
30             }
31     }
32
33     def doImports()'''
34         'import React, { Component, PropTypes } from \'react/\'
35         'import { Paper, RaisedButton, TextField }
36         from \'material-ui/\'
37         '''
38
39     def checkReactModelObject(ReactDomainModel m) '''
40         <<IF m instanceof(ReactComponent)>>
41             generateReactComponent(<<m.name>>,<<m.properties>>)
42         <<ELSEIF m instanceof(ReactMaterialUI)>>
43             generateReactMaterialUI(m)
44         <<ENDIF>>
```

## ReactDomainmodelGenerator.xtend

```

43  '''
44
45  def generateReactComponent(String ID, String props)'''
46      Component <<c.name>> render=(
47          <<IF c.instanceof(Paper)>>
48              'Paper'
49          <<ELSEIF
50      c.instanceof(Button)>>
51          'Button'
52          <<ELSEIF
53      c.instanceof(TextField)>>
54          <<ENDIF>>
55      )
56  '''
57  def generateReactMaterialUI(ReactMaterialUI m)'''
58      <<IF c.instanceof(Paper)>>
59          generateReactPaper(p)
60      <<ELSEIF c.instanceof(Button)>>
61          Button
62      <<ELSEIF c.instanceof(TextField)>>
63          <<ENDIF>>
64  '''
65
66  //mit zwei }} ??
67  def generateReactPaper(Paper p) '''
68
69      export default class MyPaper extends Component {
70          render() {
71              return (
72                  <Paper
73                      <<FOR e:p.elements>>
74                          <<IF e instanceof(StyleAttribute)>>
75                              style={
76                                  <<FOR s:e.styleItems>>
77                                      <<s.name>>:<<s.attribute>>,
78                                      <<ENDFOR>>
79                              }
80                          <<ENDIF>>
81                      <<ENDFOR>>
82                  <>
83                      <<FOR e:p.elements>>
84                          <<IF e instanceof(AddMethod)>>

```

# ReactDomainmodelGenerator.xtend

```

85         generateReactMethod(<<e.name>>)
86     <<ENDIF>>
87 <<ENDFOR>>
88 </Paper>
89 )
90 }
91 }
92 '''
93
94 //Method
95 //wieso e.name als Parameter?
96 def generateReactMethod(ReactMaterialUI m)'''
97     add('+ m +')';
98 '''
99
100 //Button
101 def generateReactButton()'''
102
103     <RaisedButton
104     <<FOR e:p.elements>>
105         <<IF e instanceof(Label)>>
106             'label="<<e.name>>"
107         <<ELSEIF>>
108             e instanceof(StyleAttribute)>>
109             style={
110                 <<FOR s:e.styleItems>>
111                     <<s.name>>:<<s.attribute>>,
112                 <<ENDIF>>
113             <<ENDFOR>> }}
114     </RaisedButton>
115
116
117
118 '''
119 //Textfield
120 def generateReactTextField()'''
121     <TextField
122     <<FOR e:p.elements>>
123         <<IF e
124             instanceof(hintText)>>
125             'hintText="<<e.name>>"
126         <<ENDIF>>
127

```

## ReactDomainmodelGenerator.xtend

```
128 <<IF e
instanceof(StyleAttribute)>>
129 style={
130 <<FOR
s:e.styleItems>>
131 <<s.name>>:<<s.attribute>>,
132 <<ENDFOR>>
133 >>}>
134
135 </TextField>
136 ' '
137
138 }
139
```