

index.js

```
1 import React from "react"
2 import ReactDOM from "react-dom"
3 import "./index.css"
4 import ManagementConsole from "./components/ManagementConsole"
5
6 ReactDOM.render(
7   <React.StrictMode>
8     <ManagementConsole/>
9   </React.StrictMode>,
10  document.getElementById("root")
11 )
```

index.css

```
1 @import '~antd/dist/antd.css';
2 /*@import '~antd/dist/antd.dark.css';*/
3
4 body {
5   margin: 0;
6   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
7   'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
8   sans-serif;
9   -webkit-font-smoothing: antialiased;
10  -moz-osx-font-smoothing: grayscale;
11 }
12
13 code {
14   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
15   monospace;
16 }
17
18 .editable-cell {
19   position: relative;
20 }
21
22 .editable-cell-value-wrap {
23   padding: 5px 12px;
24   cursor: pointer;
25 }
26
27 .editable-row:hover .editable-cell-value-wrap {
28   border: 1px solid #d9d9d9;
29   border-radius: 4px;
30   padding: 4px 11px;
31 }
32
33 [data-theme='dark'] .editable-row:hover .editable-cell-value-wrap {
34   border: 1px solid #434343;
35 }
```

EntitiesApi.js

```

1 class EntitiesApi {
2
3     static idKey = null
4     static idBuffer = null
5     //По хорошему, этот адрес должен браться извне, а не захардкоден, тем не менее, время не резиновое
6     static #rootUrl = "http://localhost:9090/"
7
8     static get(entityUrl, ids = [], isSync = false) {
9         /* Это наглядный пример плохого дизайна: изначально когда я писал апи для бекенда, я предполагал
10        , что в теле GET
11        запроса будет передаваться json-массив с id нужных элементов и пустой массив, если клиент хочет
12        получить все элементы.
13        На этапе создания фронтенда я вспомнил, что по спецификации HTTP передавать тело с GET не
14        рекомендуется, а fetch()
15        и вовсе это запрещает. Т.к. кардинально менять апи бэка не хотелось, решил костыльно передавать
16        json-массив как
17        параметр запроса. */
18     const url = this.#rootUrl + entityUrl + "?=" + new URLSearchParams({ "ids": JSON.stringify(
19         selectedIds: ids ) })
20     if (isSync) {
21         const req = new XMLHttpRequest()
22         req.open("GET", url, false)
23         req.send()
24         return req.responseText
25     } else {
26         const req = { method: "GET", mode: "cors" }
27         return fetch(url, req)
28     }
29 }
30
31 static post(entityUrl, idField, item) {
32     const req = {
33         method: "POST",
34         mode: "cors",
35         headers: { "Content-Type": "application/json" },
36         body: JSON.stringify({
37             [idField]: null,
38             ...item
39         })
40     }
41     return fetch(this.#rootUrl + entityUrl, req)
42 }
43
44 static put(entityUrl, item) {
45     const req = {
46         method: "PUT",
47         mode: "cors",
48         headers: { "Content-Type": "application/json" },
49         body: JSON.stringify(item)
50     }
51     return fetch(this.#rootUrl + entityUrl, req)
52 }
53
54 static delete(entityUrl, ids) {
55     const req = {
56         method: "DELETE",
57         mode: "cors",
58         headers: { "Content-Type": "application/json" },
59         body: JSON.stringify({ droppedIds: ids })
60     }
61     return fetch(this.#rootUrl + entityUrl, req)
62 }
63
64 export default EntitiesApi

```

ktor_logo.svg

```
1 <svg xmlns="http://www.w3.org/2000/svg" width="50" height="50" viewBox="0 0 50 50">
2   <g fill="none">
3     <path fill="#F26636" d="M16.004 0L0 16.004 16.004 32V16.004H32z"/>
4     <path fill="#6373FF" d="M48 31.996L31.996 16v15.996H16L31.996 48z"/>
5   </g>
6 </svg>
```

Credits.jsx

```
1 import React from "react"
2 import { Col, Image, Row, Divider } from "antd"
3 import KtorLogo from "../resources/images/ktor_logo.svg"
4 import ExposedLogo from "../resources/images/exposed_logo.png"
5
6 class Credits extends React.Component {
7     render() {
8         return (
9             <div style={{ textAlign: "center" }}>
10                 <Developer orientation="center">Developer:</Developer>
11                 <Row gutter={[0, 10]} align={"middle"}>
12                     <Col span={12}>
13                         <Image
14                             src="https://avatars2.githubusercontent.com/u/48172248?s=460&u=
18fe6de540ebdb9008ccf4065d7175d4ce892f7e&v=4"
15                             width={63}
16                             preview={false}/>
17                     </Col>
18                     <Col span={12}><a href="https://se.ifmo.ru/~s265570/cv/">testpassword</a></Col>
19                 </Row>
20                 <Developer orientation="center">Libraries:</Developer>
21                 <Row gutter={[0, 10]}>
22                     <Col span={12}>
23                         <Image
24                             src="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
25                             width={63}
26                             preview={false}/>
27                     </Col>
28                     <Col span={12}><a href="https://ant.design/">Antd</a></Col>
29                 </Row>
30                 <Row gutter={[0, 10]}>
31                     <Col span={12}>
32                         <Image
33                             src={KtorLogo}
34                             width={63}
35                             preview={false}/>
36                     </Col>
37                     <Col span={12}><a href="https://ktor.io/">Ktor</a></Col>
38                 </Row>
39                 <Row gutter={[0, 10]}>
40                     <Col span={12}>
41                         <Image src="https://upload.wikimedia.org/wikipedia/commons/a/a7/React-icon.svg"
42                             width={63}
43                             preview={false}/>
44                     </Col>
45                     <Col span={12}><a href="https://ru.reactjs.org/">React</a></Col>
46                 </Row>
47                 <Row gutter={[0, 10]}>
48                     <Col span={12}>
49                         <Image src={ExposedLogo}
50                             width={63}
51                             preview={false}/>
52                     </Col>
53                     <Col span={12}><a href="https://github.com/JetBrains/Exposed">Exposed</a></Col>
54                 </Row>
55                 <Row gutter={[0, 10]}>
56                     <Col span={12}>
57                         <Image
58                             src="https://upload.wikimedia.org/wikipedia/commons/2/29/Postgresql_elephant
59 .svg"
60                             width={63}
61                             preview={false}/>
62                     </Col>
63                     <Col span={12}><a href="https://www.postgresql.org/">PostgreSQL</a></Col>
64                 </Row>
65             </div>
66         )
67     }
68 }
69 export default Credits
```

EditableRow.jsx

```
1 import React, { useContext, useEffect, useRef, useState } from "react";
2 import { Form, Input } from "antd";
3
4 const EditableContext = React.createContext()
5
6 const EditableRow = ({ index, ...props }) => {
7     const [form] = Form.useForm();
8     return (
9         <Form form={form} component={false}>
10            <EditableContext.Provider value={form}>
11                <tr {...props} />
12            </EditableContext.Provider>
13        </Form>
14    )
15 }
16
17 const EditableCell = ({ title, editable, children, dataIndex, record, handleSave, ...restProps }) => {
18     const [editing, setEditing] = useState(false)
19     const inputRef = useRef()
20     const form = useContext(EditableContext);
21     useEffect(() => {
22         if (editing) inputRef.current.focus()
23     }, [editing])
24     const toggleEdit = () => {
25         setEditing(!editing)
26         form.setFieldsValue({ [dataIndex]: record[dataIndex] })
27     }
28     const save = async(e) => {
29         try {
30             const values = await form.validateFields()
31             toggleEdit()
32             handleSave({ ...record, ...values })
33         } catch (errInfo) { console.log('Save failed:', errInfo) }
34     }
35     let childNode = children
36     if (editable) {
37         childNode = editing ? (
38             <Form.Item
39                 style={{ margin: 0 }}
40                 name={ dataIndex }
41                 rules={[
42                     { required: true, message: `${title} is required.` }
43                 ]}
44             <Input ref={inputRef} onPressEnter={save} onBlur={save} />
45         </Form.Item>
46     ) : (
47         <div
48             className="editable-cell-value-wrap"
49             style={{ paddingRight: 24 }}
50             onClick={toggleEdit}>
51             {children}
52         </div>
53     )
54 }
55     return <td {...restProps}>{childNode}</td>
56 }
57
58 export { EditableRow, EditableCell }
```

EntityTable.jsx

```

1 import React from "react"
2 import {
3   Table, message, Input, Button, Space, Layout, Popover, Checkbox, InputNumber, Select, Tooltip, Modal
4   , DatePicker } from "antd"
5 import Highlighter from "react-highlight-words"
6 import { DeleteOutlined, PlusOutlined, SearchOutlined, DownloadOutlined, CloseOutlined } from "@ant-
7 design/icons"
8 import { Header, Content } from "antd/lib/layout/layout"
9 import { EditableCell, EditableRow } from "./EditableRow.jsx"
10 import { last, first, shuffle } from "underscore"
11 import EntitiesApi from "../EntitiesApi.js"
12 import EntitySimpleTable from "./EntitySimpleTable"
13 import moment from "moment"
14 import PositionPresenter from "./presentors/PositionPresenter"
15 import BasePresenter from "./presentors/BasePresenter"
16 import MrePresenter from "./presentors/MrePresenter"
17 import EmployeePresenter from "./presentors/EmployeePresenter"
18 import CampaignPresenter from "./presentors/CampaignPresenter"
19 import EquipmentPresenter from "./presentors/EquipmentPresenter"
20
21 class EntityTable extends React.Component {
22   constructor(props) {
23     super(props)
24     message.config({ duration: 5 })
25     const presenters = new Map()
26     presenters.set("posId", PositionPresenter)
27     presenters.set("baseId", BasePresenter)
28     presenters.set("mreId", MrePresenter)
29     presenters.set("empId", EmployeePresenter)
30     presenters.set("campId", CampaignPresenter)
31     presenters.set("equipId", EquipmentPresenter)
32     this.state = { additionalPresenters: presenters }
33   }
34   state = {
35     additionalPresenters: [],
36     isLoading: true,
37     items: [],
38     searchText: "",
39     searchedColumn: "",
40     selectedRowKeys: [],
41     columns: [],
42     addFormVisible: false,
43     focusedEntity: ""
44   }
45
46   getColumnSearchProps = dataIndex => ({
47     filterDropdown: ({ setSelectedKeys, selectedKeys, confirm, clearFilters }) => (
48       <div style={{ padding: 8 }}>
49         <Input
50           ref={ node => { this.searchInput = node } }
51           placeholder={`Search ${dataIndex}`}
52           value={first(selectedKeys)}
53           onChange={e => setSelectedKeys(e.target.value ? [e.target.value] : [])}
54           onPressEnter={() => this.handleSearch(selectedKeys, confirm, dataIndex)}
55           style={{ width: 188, marginBottom: 8, display: 'block' }}
56         />
57         <Space>
58           <Button
59             type="primary"
60             onClick={() => this.handleSearch(selectedKeys, confirm, dataIndex)}
61             icon={<SearchOutlined />}
62             size="small"
63             style={{ width: 90 }}>
64             Search
65           </Button>
66           <Button onClick={() => this.handleReset(clearFilters)} size="small" style={{ width:
67             90 }}>
68             Reset
69           </Button>
70         </Space>
71       </div>
72     ),
73     filterIcon: filtered => <SearchOutlined style={{ color: filtered ? '#1890ff' : undefined }} />

```

EntityTable.jsx

```

73     onFilter: (value, record) =>
74         record[dataIndex] ?
75             record[dataIndex].toString().toLowerCase().includes(value.toLowerCase())
76             : "",
77     onFilterDropdownVisibleChange: visible => { if (visible) setTimeout(() => this.searchInput.
78         select(), 100) },
79     render: text =>
80         this.state.searchedColumn === dataIndex ? (
81             <Highlighter
82                 highlightStyle={{ backgroundColor: '#ffc069', padding: 0 }}
83                 searchWords={[this.state.searchText]}
84                 autoEscape
85                 textToHighlight={text ? text.toString() : ""}
86             />
87         ) : (text)
88     })
89
90 handleSearch = (selectedKeys, confirm, dataIndex) => {
91     confirm()
92     this.setState({ searchText: first(selectedKeys), searchedColumn: dataIndex })
93 }
94
95 handleReset = clearFilters => {
96     clearFilters()
97     this.setState({ searchText: "" })
98 }
99
100 onSelectChange = selectedRowKeys => { this.setState({selectedRowKeys}) }
101
102 handleSave = modified_record => {
103     const p = this.props.presenter
104     EntitiesApi.put(p.url, modified_record)
105         .then(res => res.text())
106         .then(
107             data => {
108                 message.success({ content: data })
109                 const items = this.state.items
110                 const orig_record = items.find((it) => it[p.idField] === modified_record[p.idField]
111             ))
112                 Object.keys(modified_record).forEach(key => {
113                     if (modified_record[key] !== orig_record[key]) orig_record[key] =
114                         modified_record[key]
115                     this.setState({ items: items })
116                 },
117                 error => message.error({ content: error.message.toLocaleString() })
118             )
119 }
120
121 showConfirm = (table, onOkCallback, onCancelCallback) => {
122     const { confirm } = Modal
123     confirm({
124         zIndex: 1100,
125         width: 1100,
126         title: "Select record",
127         content: table,
128         onOk() { onOkCallback() },
129         onCancel() { if (onCancelCallback !== undefined) onCancelCallback() }
130     })
131
132 onSelectReferenceId = (key) => {
133     const editedEntityId = first(this.state.selectedRowKeys)
134     if (editedEntityId === undefined) message.error({ content: "Use should choose record before set
135     reference" })
136     else {
137         this.setState({ selectedRowKeys: [] })
138         this.handleSave({
139             [this.props.presenter.idField]: editedEntityId,
140             [key]: EntitiesApi.idBuffer
141         })
142     }
143     createColumnsFromObject = object => {

```

```

144     const p = this.props.presenter
145     const cols = Object.keys(object).map(key => {
146         const strKey = key.toString()
147         const filters = p.filteredColumns
148         let modifiedColumn = {
149             title: strKey.split(/(?=[A-Z])/).map(s => s.toUpperCase()).join(" "),
150             dataIndex: strKey,
151             defaultSortOrder: "ascend",
152             sortDirections: ["ascend", "descend"],
153             editable: (strKey !== p.idField),
154             sorter: (a, b) => (typeof a[key] === "string") ? (a[key].localeCompare(b[key])) : (a[
key] - b[key])
155         }
156         modifiedColumn = (filters !== undefined && strKey in filters) ?
157         {
158             ...modifiedColumn,
159             filters: filters[strKey],
160             onFilter: (v, r) => r[strKey] === v,
161             render: t => <Select defaultValue={t}/>
162         } :
163         {
164             ...this.getColumnSearchProps(strKey),
165             ...modifiedColumn
166         }
167         if (typeof object[key] === "boolean")
168             modifiedColumn = {
169                 ...modifiedColumn,
170                 render: t => <Checkbox checked={t}/>
171             }
172         if (typeof object[key] === "number" && modifiedColumn.editable)
173             modifiedColumn = (key.includes("Id") && key !== p.idField) ?
174             {
175                 ...modifiedColumn,
176                 editable: false,
177                 render: val =>
178                     <Tooltip title={() =>
179                         EntitiesApi.get(
180                             this.state.additionalPresenters.get(key).url,
181                             [val],
182                             true
183                         )
184                     color={ shuffle(['pink', 'red', 'yellow', 'orange', 'cyan', 'green', 'blue', 'purple', 'geekblue',
185                         'magenta', 'volcano', 'gold', 'lime'])[0] }>
186                     <Button
187                         type="link"
188                         onClick={() => this.showConfirm(
189                             <EntitySimpleTable presenter={this.state.additionalPresenters.
get(key)} />,
190                             () => this.onSelectReferenceId(key)) }>
191                         {val}
192                     </Button>
193                     </Tooltip>
194             } :
195             {
196                 ...modifiedColumn,
197                 render: t => <InputNumber defaultValue={t}/>
198             }
199         if (typeof object[key] === "object")
200             modifiedColumn = {
201                 ...modifiedColumn,
202                 render: t => {
203                     const format = "DD-MM-YYYY/HH:mm:ss"
204                     const timestamp = t.date.toString() + "/" + t.time.hour + ":" + t.time.minute
+ ":" + t.time.second
205                     console.log(timestamp)
206                     return <DatePicker showTime defaultValue={moment(timestamp, format)} format={format}>
207                         {}
208                     }
209                 if (modifiedColumn.editable)
210                     modifiedColumn = {
211                         ...modifiedColumn,
212                         onCell: (record) => ({
213                             record,

```

EntityTable.jsx

```

214             editable: modifiedColumn.editable,
215             dataIndex: modifiedColumn.dataIndex,
216             title: modifiedColumn.title,
217             handleSave: this.handleSave,
218         })
219     }
220     return modifiedColumn
221 }
222 this.setState({ columns: cols })
223 }

225 getRecords = () => {
226     EntitiesApi.get(this.props.presenter.url)
227     .then(res => {
228         this.setState({ isLoading: false })
229         return res.json()
230     })
231     .then(
232         data => {
233             this.setState({ items: data })
234             this.createColumnsFromObject(first(data))
235             message.success({ content: "Data loaded" })
236         },
237         error => message.error({ content: error.message.toLocaleString() })
238     )
239 }

240 removeRecords = () => {
241     if (this.state.selectedRowKeys.length === 0)
242         message.success({ content: "Can't delete, nothing selected" })
243     else {
244         EntitiesApi.delete(this.props.presenter.url, this.state.selectedRowKeys)
245         .then(res => res.text())
246         .then(
247             data => {
248                 message.success({ content: data })
249                 this.setState(
250                     { items: this.state.items.filter(x =>
251                         !this.state.selectedRowKeys.includes(x[this.props.presenter.idField]))
252                 })
253             },
254             error => message.error({ content: error.message.toLocaleString() })
255         )
256     }
257 }

259 addRecord = (childData) => {
260     if (childData === undefined) return
261     EntitiesApi.post(this.props.presenter.url, this.props.presenter.idField, childData).then(res
=> res.text()).then(
262         data => {
263             message.success({ content: data })
264             this.setState({ addFormVisible: false })
265             this.getRecords()
266         },
267         error => message.error({ content: error.message.toLocaleString() })
268     )
269 }

271 //Пример плохого планирования. Я попал в ад коллбеков, из-за чего на нажатие кнопки открытия формы
пришлось добавить отдельную функцию
272 handleAddClick = () => {
273     this.addRecord()
274     this.setState({ addFormVisible: true })
275 }

277 downloadRecords = () => {
278     EntitiesApi.get(
279         this.props.presenter.url,
280         (this.state.selectedRowKeys === 0) ? [] : this.state.selectedRowKeys)
281     .then(res => res.text())
282     .then(
283         data => {
284             require("file-saver").saveAs(
285                 new Blob([data], { type: "text/json;charset=utf-8" }),
286                 last(this.props.presenter.url.split("/")) + ".json"

```

EntityTable.jsx

```

287        )
288    },
289    error => message.error({ content: error.message.toLocaleString() })
290  )
291 }
292
293 // Вызывается лишь раз, при создании компонента
294 componentDidMount() {
295   this.getRecords()
296   this.updatePresenter()
297 }
298
299 // Вызывается каждый раз при обновлении props-ов из родителя
300 componentDidUpdate(prevProps) {
301   if (this.props.presenter !== prevProps.presenter) this.getRecords()
302   this.updatePresenter()
303 }
304
305 updatePresenter() {
306   this.props.presenter.creator =
307     React.cloneElement(this.props.presenter.creator, { parentCallback: this.addRecord }, null)
308 }
309
310 render() {
311   const { isLoading, items, selectedRowKeys } = this.state
312   const rowSelection = {
313     selectedRowKeys,
314     onChange: this.onSelectChange,
315     selections: [Table.SELECTION_ALL, Table.SELECTION_INVERT]
316   }
317   const components = {body: {row: EditableRow, cell: EditableCell}}
318   return <Layout className="site-layout">
319     <Header style={{ position: 'fixed', zIndex: 1, width: '100%' }}>
320       <Space size="middle">
321         <Popover
322           trigger="click"
323           content={
324             <div>
325               {this.props.presenter.creator}
326               <Button shape="round"
327                 icon={<CloseOutlined/>}
328                 size="small"
329                 onClick={() => this.setState({ addFormVisible: false })}
330               />
331             </div>
332           }
333           visible={this.state.addFormVisible}>
334             <Button type="primary"
335               icon={<PlusOutlined/>}
336               ghost={true}
337               onClick={this.handleAddClick}>
338               Add record
339             </Button>
340           </Popover>
341           <Button icon={<DeleteOutlined/>}
342             ghost={true}
343             danger
344             onClick={this.removeRecords}>
345             Remove
346           </Button>
347           <Tooltip title="All objects from table will be downloaded if nothing is selected">
348             <Button
349               icon={<DownloadOutlined/>}
350               ghost={true}
351               onClick={this.downloadRecords}>
352                 { `Download as json ${!(this.state.items === undefined) ? 0 : this.state.
      items.length}` }
353             </Button>
354           </Tooltip>
355         </Space>
356       </Header>
357       <Content style={{ marginTop: "64px" }}>
358         <div className="site-layout-background" style={{ minHeight: 360 }}>
359           <Table
360             components={components}

```

EntityTable.jsx

```
361             rowKey={this.props.presenter.idField}
362             rowSelection={rowSelection}
363             columns={this.state.columns}
364             dataSource={items}
365             loading={isLoading}
366             pagination={{ position: ["bottomCenter"] }}}
367         />
368     </div>
369   </Content>
370 </Layout>
371 }
372 }
373
374 export default EntityTable
```

EntitySimpleTable.jsx

```

1 import React from "react"
2 import EntitiesApi from "../EntitiesApi.js"
3 import {Table, message, Checkbox} from "antd"
4
5 class EntitySimpleTable extends React.Component {
6
7     state = { items: [] }
8
9     createSimpleColumnsFromObject = object => {
10         const cols = Object.keys(object).map(key => {
11             const strKey = key.toString()
12             let modifiedColumn = {
13                 title: strKey.split(/(?=[A-Z])/).map(s => s.toUpperCase()).join(" "),
14                 dataIndex: strKey,
15                 defaultSortOrder: "ascend",
16                 sortDirections: ["ascend", "descend"],
17                 sorter: (a, b) => a[key].localeCompare(b[key])
18             }
19             if (typeof object[key] === "boolean")
20                 modifiedColumn = {
21                     ...modifiedColumn,
22                     render: t => <Checkbox checked={t}/>
23                 }
24             return modifiedColumn
25         })
26         this.setState({ columns: cols })
27     }
28
29     getRecords = () => {
30         EntitiesApi.get(this.props.presenter.url)
31             .then(res => res.json())
32             .then(
33                 data => {
34                     this.setState({ items: data })
35                     this.createSimpleColumnsFromObject(data[0])
36                 },
37                 error => message.error({ content: error.message.toLocaleString() })
38             )
39     }
40
41     componentDidMount() { this.getRecords() }
42
43     /* Здесь я не придумал, как передать ключ выбранного элемента на два родителя вверх, поэтому просто
отправил его
        в глобальную переменную. ПОВТОРЯТЬ ТАКОЕ НЕЛЬЗЯ! */
44     rowSelection = {
45         onChange: (selectedRowKeys) => {
46             EntitiesApi.idKey = this.props.presenter.idField
47             EntitiesApi.idBuffer = selectedRowKeys[0]
48         }
49     }
50
51
52     render() {
53         return <Table
54             scroll={{ x: 800 }}
55             rowSelection={{ type: 'radio', ...this.rowSelection }}
56             rowKey={this.props.presenter.idField}
57             columns={this.state.columns}
58             dataSource={this.state.items}
59         />
60     }
61 }
62
63 export default EntitySimpleTable

```

ManagementConsole.jsx

```

1 import React from "react"
2 import { Layout, Menu, Modal } from "antd"
3 import Credits from "./Credits"
4 import { CalendarOutlined, TeamOutlined, CarOutlined, HomeOutlined, GlobalOutlined, MedicineBoxOutlined,
5   ShoppingOutlined,
6   MoneyCollectOutlined, CoffeeOutlined } from "@ant-design/icons"
7 import EntityTable from "./EntityTable"
8 import BasePresenter from "./presentors/BasePresenter.jsx"
9 import CampaignPresenter from "./presentors/CampaignPresenter.jsx"
10 import EmployeePresenter from "./presentors/EmployeePresenter.jsx"
11 import EquipmentPresenter from "./presentors/EquipmentPresenter"
12 import MedicalCardPresenter from "./presentors/MedicalCardPresenter"
13 import MissionPresenter from "./presentors/MissionPresenter"
14 import MrePresenter from "./presentors/MrePresenter"
15 import PositionPresenter from "./presentors/PositionPresenter"
16 import TransportPresenter from "./presentors/TransportPresenter"
17 import WeaponPresenter from "./presentors/WeaponPresenter"
18
19 class ManagementConsole extends React.Component {
20
21   state = {
22     menuIsCollapsed: false,
23     presenter: BasePresenter
24   }
25
26   onCollapse = menuIsCollapsed => { this.setState({ menuIsCollapsed }) }
27
28   changeEntityTable(p) { this.setState({ presenter: p }) }
29
30   info() {
31     Modal.info({
32       title: "Credits",
33       content: <Credits/>
34     })
35   }
36
37   render() {
38     const { menuIsCollapsed } = this.state;
39     const { Content, Footer, Sider } = Layout;
40     return (
41       <Layout style={{ minHeight: '100vh' }} theme="dark">
42         <Sider collapsible collapsed={menuIsCollapsed} onCollapse={this.onCollapse}>
43           <Menu theme="dark" defaultSelectedKeys={['1']} mode="inline">
44             <Menu.Item
45               icon={<HomeOutlined/>}
46               onClick={() => this.changeEntityTable(BasePresenter)}>
47               Bases
48             </Menu.Item>
49             <Menu.Item
50               icon={<GlobalOutlined/>}
51               onClick={() => this.changeEntityTable(CampaignPresenter)}>
52               Campaigns
53             </Menu.Item>
54             <Menu.Item
55               icon={<TeamOutlined/>}
56               onClick={() => this.changeEntityTable(EmployeePresenter)}>
57               Employees
58             </Menu.Item>
59             <Menu.Item
60               icon={<ShoppingOutlined/>}
61               onClick={() => this.changeEntityTable(EquipmentPresenter)}>
62               Equipments
63             </Menu.Item>
64             <Menu.Item
65               icon={<MedicineBoxOutlined/>}
66               onClick={() => this.changeEntityTable(MedicalCardPresenter)}>
67               Medical Cards
68             </Menu.Item>
69             <Menu.Item
70               icon={<CalendarOutlined/>}
71               onClick={() => this.changeEntityTable(MissionPresenter)}>
72               Missions
73             </Menu.Item>
74             <Menu.Item
75               icon={<CoffeeOutlined/>}
```

ManagementConsole.jsx

```
75          onClick={() => this.changeEntityTable(MrePresenter)}>
76          MREs
77      </MenuItem>
78      <MenuItem
79          icon={<MoneyCollectOutlined/>}
80          onClick={() => this.changeEntityTable(PositionPresenter)}>
81          Positions
82      </MenuItem>
83      <MenuItem
84          icon={<CarOutlined/>}
85          onClick={() => this.changeEntityTable(TransportPresenter)}>
86          Transports
87      </MenuItem>
88      <MenuItem
89          onClick={() => this.changeEntityTable(WeaponPresenter)}>
90          Weapons
91      </MenuItem>
92      </Menu>
93  </Sider>
94  <Layout className="site-layout">
95      <Content>
96          <div className="site-layout-background" style={{ minHeight: 360 }}>
97              <EntityTable presenter={this.state.presenter}/>
98          </div>
99      </Content>
100     <Footer style={{ textAlign: "center" }}>
101         <a onClick={ this.info }>Kulbako Artemy ©2020</a>
102     </Footer>
103  </Layout>
104 </Layout>
105 )
106 }
107 }
108
109 export default ManagementConsole
```

MrePresenter.jsx

```
1 import React from "react"
2 import {Button, Form, Input, InputNumber} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4
5 class MreCreator extends AbstractCreator {
6
7     render() {
8         return <Form onFinish={this.onTrigger}>
9             <Form.Item label="Breakfast"
10                 name="breakfast"
11                 rules={[{ required: true, message: "Set breakfast! Hungry solder - dead solder." }]}>
12                 <Input/>
13             </Form.Item>
14             <Form.Item label="Carbohydrate"
15                 name="carbohydrate"
16                 rules={[{ required: true, message: "Set total carbohydrate value in MRE" }]}>
17                 <InputNumber min={0}/>
18             </Form.Item>
19             <Form.Item label="Dinner"
20                 name="dinner"
21                 rules={[{ required: true, message: "Set dinner! Hungry solder - dead solder." }]}>
22                 <Input/>
23             </Form.Item>
24             <Form.Item label="Fats"
25                 name="fats"
26                 rules={[{ required: true, message: "Set total fats value in MRE" }]}>
27                 <InputNumber min={0}/>
28             </Form.Item>
29             <Form.Item label="Food additives"
30                 name="foodAdditives">
31                 <Input placeholder="Use space as separator"/>
32             </Form.Item>
33             <Form.Item label="Kkal"
34                 name="kkal"
35                 rules={[{ required: true, message: "Set total kkal value in MRE" }]}>
36                 <InputNumber min={3000}/>
37             </Form.Item>
38             <Form.Item label="Lunch"
39                 name="lunch"
40                 rules={[{ required: true, message: "Set lunch! Hungry solder - dead solder." }]}>
41                 <Input/>
42             </Form.Item>
43             <Form.Item label="Proteins"
44                 name="proteins"
45                 rules={[{ required: true, message: "Set total proteins value in MRE" }]}>
46                 <Input min={0}/>
47             </Form.Item>
48             <Form.Item>
49                 <Button type="primary" htmlType="submit">Submit</Button>
50             </Form.Item>
51         </Form>
52     }
53 }
54
55 const MrePresenter = {
56     url: "mre",
57     idField: "mreId",
58     creator: <MreCreator/>
59 }
60
61 export default MrePresenter
```

BasePresenter.jsx

```
1 import React from "react"
2 import { Input, Form, Button, Radio } from "antd"
3 import AbstractCreator from "./AbstractCreator"
4
5 class BaseCreator extends AbstractCreator {
6
7     render() {
8         return <Form onFinish={this.onTrigger}>
9             <Form.Item label="Location"
10                 name="location"
11                 rules={[{ required: true, message: "Input base location" }]}>
12                 <Input/>
13             </Form.Item>
14             <Form.Item label="Status"
15                 name="status"
16                 rules={[{ required: true, message: "Choose base status" }]}>
17                 <Radio.Group
18                     options={BasePresenter.filteredColumns.status.map(o => o.text)}
19                     optionType="button"
20                 />
21             </Form.Item>
22             <Form.Item>
23                 <Button type="primary" htmlType="submit">Submit</Button>
24             </Form.Item>
25         </Form>
26     }
27 }
28
29 const BasePresenter = {
30     url: "base",
31     idField: "baseId",
32     filteredColumns: {
33         status: [
34             { text: "working", value: "working" },
35             { text: "closed", value: "closed" },
36             { text: "destroyed", value: "destroyed" },
37             { text: "abandoned", value: "abandoned" },
38             { text: "captured", value: "captured" },
39             { text: "for_sale", value: "for_sale" }
40         ]
41     },
42     creator: <BaseCreator/>
43 }
44
45 export default BasePresenter
```

AbstractCreator.jsx

```
1 import React from "react"
2 import { Modal } from "antd"
3
4 class AbstractCreator extends React.Component {
5
6     onTrigger = (formData) => this.props.parentCallback(formData)
7
8     showConfirm = (table, onOkCallback, onCancelCallback) => {
9         const { confirm } = Modal
10        confirm({
11            zIndex: 1100,
12            width: 1100,
13            title: "Select record",
14            content: table,
15            onOk() { onOkCallback() },
16            onCancel() { if (onCancelCallback !== undefined) onCancelCallback() }
17        })
18    }
19 }
20
21 export default AbstractCreator
```

WeaponPresenter.jsx

```
1 import React from "react"
2 import {Button, Form, Input, InputNumber} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4
5 class WeaponCreator extends AbstractCreator {
6
7     render() {
8         return <Form onFinish={this.onTrigger}>
9             <Form.Item label="Caliber"
10                 name="caliber">
11                 <InputNumber min={0}/>
12             </Form.Item>
13             <Form.Item label="Name"
14                 name="name"
15                 rules={[{ required: true, message: "Input name of weapon" }]}>
16                 <Input/>
17             </Form.Item>
18             <Form.Item label="Rate of fire"
19                 name="rateOfFire">
20                 <InputNumber min={0}/>
21             </Form.Item>
22             <Form.Item label="Sighting range m"
23                 name="sightingRangeM">
24                 <InputNumber min={0}/>
25             </Form.Item>
26             <Form.Item label="Type"
27                 name="type"
28                 rules={[{ required: true, message: "Input type of weapon" }]}>
29                 <Input/>
30             </Form.Item>
31             <Form.Item>
32                 <Button type="primary" htmlType="submit">Submit</Button>
33             </Form.Item>
34         </Form>
35     }
36 }
37
38 const WeaponPresenter = {
39     url: "weapon",
40     idField: "weaponId",
41     creator: <WeaponCreator/>
42 }
43
44 export default WeaponPresenter
```

MissionPresenter.jsx

```

1 import React from "react"
2 import {Button, DatePicker, Form, Input, message, Radio} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4 import EntitySimpleTable from "../EntitySimpleTable"
5 import CampaignPresenter from "./CampaignPresenter"
6 import EntitiesApi from "../../EntitiesApi"
7
8 class MissionCreator extends AbstractCreator {
9
10    state = { campBtn: "select campaign" }
11
12    campOk = () => {
13        if (EntitiesApi.idBuffer !== null) {
14            this.setState({ campBtn: EntitiesApi.idBuffer })
15        } else message.error({ content: "You should choose base from table" })
16    }
17
18    onTrigger = (formData) => {
19        this.props.parentCallback({
20            ...formData,
21            campId: this.state.campBtn
22        })
23    }
24
25    render() {
26        const { RangePicker } = DatePicker
27        return <Form onFinish={this.onTrigger}>
28            <Form.Item label="Arrival location"
29                name="arrivalLocation">
30                <Input placeholder="Use space as separator"/>
31            </Form.Item>
32            <Form.Item label="Campaign ID"
33                name="campId">
34                <Button type="link"
35                    onClick={() => this.showConfirm(<EntitySimpleTable presenter={CampaignPresenter
}/>, this.campOk) }>
36                    {this.state.campBtn}
37                </Button>
38            </Form.Item>
39            <Form.Item label="Departure location"
40                name="departureLocation">
41                <Input placeholder="Use space as separator"/>
42            </Form.Item>
43            <Form.Item label="Period"
44                name="period">
45                <RangePicker showTime/>
46            </Form.Item>
47            <Form.Item label="Enemies"
48                name="enemies">
49                <Input/>
50            </Form.Item>
51            <Form.Item label="Legal status"
52                name="legalStatus"
53                rules={[{ required: true, message: "Choose legality status of mission" }]}>
54                <Radio.Group>
55                    <Radio value={true}>legal</Radio>
56                    <Radio value={false}>illegal</Radio>
57                </Radio.Group>
58            </Form.Item>
59            <Form.Item>
60                <Button type="primary" htmlType="submit">Submit</Button>
61            </Form.Item>
62        </Form>
63    }
64 }
65
66 const MissionPresenter = {
67     url: "mission",
68     idField: "missId",
69     filteredColumns: {
70         legalStatus: [
71             { text: "legal", value: true },
72             { text: "illegal", value: false }
73         ]
74     },

```

MissionPresenter.jsx

```
75     creator: <MissionCreator/>
76 }
77
78 export default MissionPresenter
```

CampaignPresenter.jsx

```
1 import React from "react"
2 import { Button, Form, Input, InputNumber, Radio } from "antd"
3 import AbstractCreator from "./AbstractCreator"
4
5 class CampaignCreator extends AbstractCreator {
6
7     render() {
8         return <Form onFinish={this.onTrigger}>
9             <Form.Item label="Customer"
10                 name="customer"
11                 rules={[{ required: true, message: "Input customer name" }]}>
12                 <Input/>
13             </Form.Item>
14             <Form.Item label="Earning"
15                 name="earning"
16                 rules={[{ required: true, message: "Input earning for campaign" }]}>
17                 <InputNumber
18                     formatter={ v => `\$ ${v}`.replace(/\B(?=(\d{3})+(?!\d))/g, ',') }
19                     parser={ v => v.replace(/\$s?|,/g, '') }/>>
20             </Form.Item>
21             <Form.Item label="Execution status"
22                 name="executionStatus"
23                 rules={[{ required: true, message: "Execution status of campaign" }]}>
24                 <Radio.Group
25                     options={CampaignPresenter.filteredColumns.executionStatus.map(o => o.text)}
26                     optionType="button"
27                 />
28             </Form.Item>
29             <Form.Item label="Name"
30                 name="name"
31                 rules={[{ required: true, message: "Input of campaign" }]}>
32                 <Input/>
33             </Form.Item>
34             <Form.Item label="Spending"
35                 name="spending"
36                 rules={[{ required: true, message: "Spending for campaign" }]}>
37                 <InputNumber
38                     formatter={ v => `\$ ${v}`.replace(/\B(?=(\d{3})+(?!\d))/g, ',') }
39                     parser={ v => v.replace(/\$s?|,/g, '') }/>>
40             </Form.Item>
41             <Form.Item>
42                 <Button type="primary" htmlType="submit">Submit</Button>
43             </Form.Item>
44         </Form>
45     }
46 }
47
48 const CampaignPresenter = {
49     url: "campaign",
50     idField: "campId",
51     filteredColumns: {
52         executionStatus: [
53             { text: "completed", value: "completed" },
54             { text: "in the progress", value: "in the progress" },
55             { text: "failed", value: "failed" },
56             { text: "canceled", value: "canceled" }
57         ]
58     },
59     creator: <CampaignCreator/>
60 }
61
62 export default CampaignPresenter
```

EmployeePresenter.jsx

```

1 import React from "react"
2 import {Button, DatePicker, Form, Input, Radio, message} from "antd"
3 import moment from "moment"
4 import EntitySimpleTable from "../EntitySimpleTable"
5 import AbstractCreator from "./AbstractCreator"
6 import EntitiesApi from "../../EntitiesApi"
7 import BasePresenter from "./BasePresenter"
8 import PositionPresenter from "./PositionPresenter"
9
10 class EmployeeCreator extends AbstractCreator {
11
12     state = {
13         baseBtn: "select base",
14         posBtn: "select position",
15     }
16
17     baseOk = () => {
18         if (EntitiesApi.idBuffer !== null) {
19             this.setState({ baseBtn: EntitiesApi.idBuffer })
20         } else message.error({ content: "You should choose base from table" })
21     }
22
23     posOk = () => {
24         if (EntitiesApi.idBuffer !== null) {
25             this.setState({ posBtn: EntitiesApi.idBuffer })
26         } else message.error({ content: "You should choose employee position from table" })
27     }
28
29     onTrigger = (formData) =>
30         this.props.parentCallback({
31             ...formData,
32             posId: this.state.posBtn,
33             baseId: this.state.baseBtn
34         })
35
36     render() {
37         return <Form onFinish={this.onTrigger}>
38             <Form.Item label="Base ID"
39                 name="baseId">
40                 <Button type="link"
41                     onClick={() => this.showConfirm(<EntitySimpleTable presenter={BasePresenter}/>
, this.baseOk)}>
42                     {this.state.baseBtn}
43                 </Button>
44             </Form.Item>
45             <Form.Item label="Date of birth"
46                 name="dateOfBirth"
47                 rules={[{ required: true, message: "Input date of birth new employee" }]}>
48                 <DatePicker format={['DD/MM/YYYY', 'DD/MM/YY']} />
49             </Form.Item>
50             <Form.Item label="Education"
51                 name="education"
52                 rules={[{ message: "Input education of new employee" }]}>
53                 <Input/>
54             </Form.Item>
55             <Form.Item label="Hiring date"
56                 name="hiringDate"
57                 rules={[{ required: true, message: "Input date of hiring new employee" }]}>
58                 <DatePicker format="DD.MM.YYYY" defaultValue={moment()} />
59             </Form.Item>
60             <Form.Item label="Is married"
61                 name="isMarried"
62                 rules={[{ required: true, message: "Choose married status of new employee" }]}>
63                 <Radio.Group>
64                     <Radio value={true}>yes</Radio>
65                     <Radio value={false}>no</Radio>
66                 </Radio.Group>
67             </Form.Item>
68             <Form.Item label="Name"
69                 name="name"
70                 rules={[{ required: true, message: "Input employee name" }]}>
71                 <Input/>
72             </Form.Item>
73             <Form.Item label="Position ID"
74                 name="posId">

```

EmployeePresenter.jsx

```
75             <Button type="link"
76                 onClick={() => this.showConfirm(<EntitySimpleTable presenter={
77                   PositionPresenter}>/), this.posOk} >
77               {this.state.posBtn}
78             </Button>
79           </Form.Item>
80           <Form.Item label="Surname"
81             name="surname"
82             rules={[{ required: true, message: "Input employee surname" }]}>
83             <Input/>
84           </Form.Item>
85           <Form.Item>
86             <Button type="primary" htmlType="submit">Submit</Button>
87           </Form.Item>
88         </Form>
89       }
90     }
91
92 const EmployeePresenter = {
93   url: "employee",
94   idField: "empId",
95   filteredColumns: {
96     isMarried: [
97       { text: "married", value: true },
98       { text: "single", value: false }
99     ]
100   },
101   creator: <EmployeeCreator/>
102 }
103
104 export default EmployeePresenter
```

PositionPresenter.jsx

```

1 import React from "react"
2 import {Button, Form, Input, InputNumber, message, Radio} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4 import EntitiesApi from "../../EntitiesApi"
5 import EntitySimpleTable from "../EntitySimpleTable"
6 import EquipmentPresenter from "./EquipmentPresenter"
7
8 class PositionCreator extends AbstractCreator {
9
10    state = { equipBtn: "select position" }
11
12    equipOk = () => {
13        if (EntitiesApi.idBuffer !== null) {
14            this.setState({ equipBtn: EntitiesApi.idBuffer })
15        } else message.error({ content: "You should choose equipment from table" })
16    }
17
18    onTrigger = (formData) => {
19        this.props.parentCallback({
20            ...formData,
21            equipId: this.state.equipBtn
22        })
23    }
24
25    render() {
26        return <Form onFinish={this.onTrigger}>
27            <Form.Item label="Equipment ID"
28                name="equipId">
29                <Button type="link"
30                    onClick={() => this.showConfirm(<EntitySimpleTable presenter={EquipmentPresenter}/>, this.equipOk)}>
31                    {this.state.equipBtn}
32                </Button>
33            </Form.Item>
34            <Form.Item label="Forces"
35                name="forces">
36                <Radio.Group
37                    options={PositionPresenter.filteredColumns.forces.map(o => o.text)}
38                    optionType="button"
39                />
40            </Form.Item>
41            <Form.Item label="name"
42                name="name"
43                rules={[{ required: true, message: "Input position name" }]}>
44                <Input/>
45            </Form.Item>
46            <Form.Item label="Rank"
47                name="rank">
48                <Input/>
49            </Form.Item>
50            <Form.Item label="Salary"
51                name="salary"
52                rules={[{ required: true, message: "Set employee salary - it should be greater
then three hundred bucks" }]}>
53                <InputNumber min={300}/>
54            </Form.Item>
55            <Form.Item>
56                <Button type="primary" htmlType="submit">Submit</Button>
57            </Form.Item>
58        </Form>
59    }
60 }
61
62 const PositionPresenter = {
63     url: "position",
64     idField: "posId",
65     filteredColumns: {
66         forces: [
67             { text: "NAVY", value: "NAVY" },
68             { text: "AF", value: "AF" },
69             { text: "GF", value: "GF" }
70         ]
71     },
72     creator: <PositionCreator/>
73 }

```

PositionPresenter.jsx

```
74  
75 export default PositionPresenter
```

EquipmentPresenter.jsx

```

1 import React from "react"
2 import {Button, Form, Input, message} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4 import EntitySimpleTable from "../EntitySimpleTable"
5 import MrePresenter from "./MrePresenter"
6 import EntitiesApi from "../../EntitiesApi"
7
8 class EquipmentCreator extends AbstractCreator {
9
10    state = { mreBtn: "select mre" }
11
12    mreOk = () => {
13        if (EntitiesApi.idBuffer !== null) {
14            this.setState({ mreBtn: EntitiesApi.idBuffer })
15        } else message.error({ content: "You should choose mre from table" })
16    }
17
18    onTrigger = (formData) => {
19        this.props.parentCallback({
20            ...formData,
21            mreId: this.state.mreBtn
22        })
23    }
24
25    render() {
26        return <Form onFinish={this.onTrigger}>
27            <Form.Item label="Camouflage"
28                name="camouflage">
29                <Input/>
30            </Form.Item>
31            <Form.Item label="Communication"
32                name="communication">
33                <Input/>
34            </Form.Item>
35            <Form.Item label="Intelligence"
36                name="intelligence">
37                <Input/>
38            </Form.Item>
39            <Form.Item label="Medical"
40                name="medical">
41                <Input/>
42            </Form.Item>
43            <Form.Item label="Extras"
44                name="extra">
45                <Input placeholder="Use space as separator"/>
46            </Form.Item>
47            <Form.Item label="Mre ID"
48                name="mreId">
49                <Button type="link"
50                    onClick={ () => this.showConfirm(<EntitySimpleTable presenter={MrePresenter}/>,
this.mreOk) }>
51                    {this.state.mreBtn}
52                </Button>
53            </Form.Item>
54            <Form.Item>
55                <Button type="primary" htmlType="submit">Submit</Button>
56            </Form.Item>
57        </Form>
58    }
59 }
60
61 const EquipmentPresenter = {
62     url: "equipment",
63     idField: "equipId",
64     creator: <EquipmentCreator/>
65 }
66
67 export default EquipmentPresenter

```

TransportPresenter.jsx

```
1 import React from "react"
2 import {Button, Form, Input, Radio} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4
5 class TransportCreator extends AbstractCreator {
6
7     render() {
8         return <Form onFinish={this.onTrigger}>
9             <Form.Item label="Name"
10                 name="name"
11                 rules={[{ required: true, message: "Input transport name" }]}>
12                 <Input/>
13             </Form.Item>
14             <Form.Item label="Status"
15                 name="status"
16                 rules={[{ required: true, message: "Choose status of transport" }]}>
17                 <Radio.Group
18                     options={TransportPresenter.filteredColumns.status.map(o => o.text)}
19                     optionType="button"
20                 />
21             </Form.Item>
22             <Form.Item label="Type"
23                 name="type"
24                 rules={[{ required: true, message: "Input type of transport" }]}>
25                 <Input/>
26             </Form.Item>
27             <Form.Item>
28                 <Button type="primary" htmlType="submit">Submit</Button>
29             </Form.Item>
30         </Form>
31     }
32 }
33
34 const TransportPresenter = {
35     url: "transport",
36     idField: "transId",
37     filteredColumns: {
38         status: [
39             { text: "available", value: "available" },
40             { text: "under_repair", value: "under_repair" },
41             { text: "destroyed", value: "destroyed" },
42             { text: "broken", value: "broken" }
43         ]
44     },
45     creator: <TransportCreator/>
46 }
47
48 export default TransportPresenter
```

MedicalCardPresenter.jsx

```

1 import React from "react"
2 import {Button, Form, Input, InputNumber, message, Radio} from "antd"
3 import AbstractCreator from "./AbstractCreator"
4 import EntitySimpleTable from "../EntitySimpleTable"
5 import EmployeePresenter from "./EmployeePresenter"
6 import EntitiesApi from "../../EntitiesApi"
7
8 class MedicalCardCreator extends AbstractCreator {
9
10    state = { empBtn: "select employee" }
11
12    empOk = () => {
13        if (EntitiesApi.idBuffer !== null) {
14            this.setState({ empBtn: EntitiesApi.idBuffer })
15        } else message.error({ content: "You should choose employee from table" })
16    }
17
18    onTrigger = (formData) => {
19        this.props.parentCallback({
20            ...formData,
21            empId: this.state.empBtn
22        })
23    }
24
25    render() {
26        return <Form onFinish={this.onTrigger}>
27            <Form.Item label="Diseases"
28                name="diseases">
29                <Input placeholder="Use space as separator"/>
30            </Form.Item>
31            <Form.Item label="Gender"
32                name="gender"
33                rules={[{ required: true, message: "Choose gender" }]}>
34                <Radio.Group>
35                    <Radio value={true}>male</Radio>
36                    <Radio value={false}>female</Radio>
37                </Radio.Group>
38            </Form.Item>
39            <Form.Item label="Height cm"
40                name="heightCm"
41                rules={[{ required: true, message: "Input employee height" }]}>
42                <InputNumber/>
43            </Form.Item>
44            <Form.Item label="Weight kg"
45                name="weightKg"
46                rules={[{ required: true, message: "Input employee weight" }]}>
47                <InputNumber/>
48            </Form.Item>
49            <Form.Item label="Blood"
50                name="blood"
51                rules={[{ required: true, message: "Choose blood type - it's about saving life!" }]}>
52                <Radio.Group
53                    options={MedicalCardPresenter.filteredColumns.blood.map(o => o.text)}
54                    optionType="button"
55                />
56            </Form.Item>
57            <Form.Item label="Employee ID"
58                name="empId">
59                <Button type="link"
60                    onClick={() => this.showConfirm(<EntitySimpleTable presenter={EmployeePresenter
61                }/>, this.empOk) }>
62                    {this.state.empBtn}
63                </Button>
64            </Form.Item>
65            <Form.Item>
66                <Button type="primary" htmlType="submit">Submit</Button>
67            </Form.Item>
68        </Form>
69    }
70
71 const MedicalCardPresenter = {
72     url: "medicalCard",
73     idField: "medId",

```

MedicalCardPresenter.jsx

```
74     filteredColumns: {
75       blood: [
76         { text: "0-", value: "0-" },
77         { text: "0+", value: "0+" },
78         { text: "A-", value: "A-" },
79         { text: "A+", value: "A+" },
80         { text: "B-", value: "B-" },
81         { text: "AB-", value: "AB-" },
82         { text: "AB+", value: "AB+" },
83       ],
84       gender: [
85         { text: "male", value: true },
86         { text: "female", value: false }
87       ]
88     },
89     creator: <MedicalCardCreator/>
90   }
91
92 export default MedicalCardPresenter
```