

```

import * as React from 'react';
 2 import { Admin } from 'react-admin';
 3 import { Resource, required, useCreate, useRecordContext } from 'ra-core';
 4 import { createMemoryHistory } from 'history';
 5 import {
 6     Dialog,
 7     DialogContent,
 8     TextField,
 9     DialogActions,
10     Button,
11     Stack,
12 } from '@mui/material';
13
14 import { Edit } from '../detail';
15 import { SimpleForm } from '../form';
16 import { AutocompleteInput } from './AutocompleteInput';
17 import { ReferenceInput } from './ReferenceInput';
18 import { useCreateSuggestionContext } from './useSupportCreateSuggestion';
19
20 export default { title: 'ra-ui-materialui/input/AutocompleteInput' };
21
22 const dataProvider = {
23     getOne: (resource, params) =>
24         Promise.resolve({
25             data: {
26                 id: 1,
27                 title: 'War and Peace',
28                 author: 1,
29                 summary:
30                     "War and Peace broadly focuses on Napoleon's invasion of
Russia, and the impact it had on Tsarist society. The book explores themes such as
revolution, revolution and empire, the growth and decline of various states and the
impact it had on their economies, culture, and society.",
31                 year: 1869,
32             },
33         }),
34     update: (resource, params) => Promise.resolve(params),
35 } as any;
36
37 const history = createMemoryHistory({ initialEntries: ['/books/1'] });
38
39 const BookEdit = () => {
40     const choices = [
41         { id: 1, name: 'Leo Tolstoy' },
42         { id: 2, name: 'Victor Hugo' },
43         { id: 3, name: 'William Shakespeare' },
44         { id: 4, name: 'Charles Baudelaire' },
45         { id: 5, name: 'Marcel Proust' },
46     ];
47     return (
48         <Edit
49             mutationMode="pessimistic"
50             mutationOptions={{
51                 onSuccess: data => {
52                     console.log(data);
53                 },
54             }}
55         >
56             <SimpleForm>
57                 <AutocompleteInput
58                     source="author"
59                     choices={choices}
60                     validate={required()}
61                     fullWidth
62                 />
63             </SimpleForm>
64         </Edit>
65     );

```

```

66 };
67
68 export const Basic = () => (
69   <Admin dataProvider={dataProvider} history={history}>
70     <Resource name="books" edit={BookEdit} />
71   </Admin>
72 );
73
74 const BookEditCustomText = () => {
75   const choices = [
76     { id: 1, fullName: 'Leo Tolstoy' },
77     { id: 2, fullName: 'Victor Hugo' },
78     { id: 3, fullName: 'William Shakespeare' },
79     { id: 4, fullName: 'Charles Baudelaire' },
80     { id: 5, fullName: 'Marcel Proust' },
81   ];
82   return (
83     <Edit
84       mutationMode="pessimistic"
85       mutationOptions={{
86         onSuccess: data => {
87           console.log(data);
88         },
89       }}
90     >
91       <SimpleForm>
92         <AutocompleteInput
93           source="author"
94           optionText="fullName"
95           choices={choices}
96           fullWidth
97         />
98       </SimpleForm>
99     </Edit>
100   );
101 };
102
103 export const CustomText = () => (
104   <Admin dataProvider={dataProvider} history={history}>
105     <Resource name="books" edit={BookEditCustomText} />
106   </Admin>
107 );
108
109 const BookEditCustomTextFunction = () => {
110   const choices = [
111     { id: 1, fullName: 'Leo Tolstoy' },
112     { id: 2, fullName: 'Victor Hugo' },
113     { id: 3, fullName: 'William Shakespeare' },
114     { id: 4, fullName: 'Charles Baudelaire' },
115     { id: 5, fullName: 'Marcel Proust' },
116   ];
117   return (
118     <Edit
119       mutationMode="pessimistic"
120       mutationOptions={{
121         onSuccess: data => {
122           console.log(data);
123         },
124       }}
125     >
126       <SimpleForm>
127         <AutocompleteInput
128           source="author"
129           optionText={choice => choice?.fullName}
130           choices={choices}
131           fullWidth
132         />
133       </SimpleForm>
134     </Edit>

```

```

135     );
136 };
137
138 export const CustomTextFunction = () => (
139   <Admin dataProvider={dataProvider} history={history}>
140     <Resource name="books" edit={BookEditCustomTextFunction} />
141   </Admin>
142 );
143
144 const CustomOption = props => {
145   const record = useRecordContext();
146
147   return (
148     <div {...props}>
149       {record?.fullName}&nbsp;<i>({record?.language})</i>
150     </div>
151   );
152 };
153
154 const BookEditCustomOptions = () => {
155   const choices = [
156     { id: 1, fullName: 'Leo Tolstoy', language: 'Russian' },
157     { id: 2, fullName: 'Victor Hugo', language: 'French' },
158     { id: 3, fullName: 'William Shakespeare', language: 'English' },
159     { id: 4, fullName: 'Charles Baudelaire', language: 'French' },
160     { id: 5, fullName: 'Marcel Proust', language: 'French' },
161   ];
162   return (
163     <Edit
164       mutationMode="pessimistic"
165       mutationOptions={{
166         onSuccess: data => {
167           console.log(data);
168         },
169       }}
170     >
171       <SimpleForm>
172         <AutocompleteInput
173           fullWidth
174           source="author"
175           optionText={<CustomOption />}
176           inputText={record => record.fullName}
177           matchSuggestion={(searchText, record) => {
178             const searchTextLower = searchText.toLowerCase();
179             const match =
180               record.fullName
181                 .toLowerCase()
182                 .includes(searchTextLower) ||
183               record.language
184                 .toLowerCase()
185                 .includes(searchTextLower);
186
187             return match;
188           }}
189           choices={choices}
190         />
191       </SimpleForm>
192     </Edit>
193   );
194 };
195
196 export const CustomOptions = () => (
197   <Admin dataProvider={dataProvider} history={history}>
198     <Resource name="books" edit={BookEditCustomOptions} />
199   </Admin>
200 );
201
202 const choicesForCreationSupport = [
203   { id: 1, name: 'Leo Tolstoy' },

```

```

204     { id: 2, name: 'Victor Hugo' },
205     { id: 3, name: 'William Shakespeare' },
206     { id: 4, name: 'Charles Baudelaire' },
207     { id: 5, name: 'Marcel Proust' },
208 ];
209 const BookEditWithCreationSupport = () => (
210   <Edit
211     mutationMode="pessimistic"
212     mutationOptions={{
213       onSuccess: data => {
214         console.log(data);
215       },
216     }}
217   >
218     <SimpleForm>
219       <AutocompleteInput
220         source="author"
221         choices={choicesForCreationSupport}
222         onCreate={filter => {
223           const newAuthorName = window.prompt(
224             'Enter a new author',
225             filter
226           );
227
228           if (newAuthorName) {
229             const newAuthor = {
230               id: choicesForCreationSupport.length + 1,
231               name: newAuthorName,
232             };
233             choicesForCreationSupport.push(newAuthor);
234             return newAuthor;
235           }
236         }}
237         fullWidth
238       />
239     </SimpleForm>
240   </Edit>
241 );
242
243 export const CreationSupport = () => (
244   <Admin dataProvider={dataProvider} history={history}>
245     <Resource name="books" edit={BookEditWithCreationSupport} />
246   </Admin>
247 );
248
249 const authors = [
250   { id: 1, name: 'Leo Tolstoy', language: 'Russian' },
251   { id: 2, name: 'Victor Hugo', language: 'French' },
252   { id: 3, name: 'William Shakespeare', language: 'English' },
253   { id: 4, name: 'Charles Baudelaire', language: 'French' },
254   { id: 5, name: 'Marcel Proust', language: 'French' },
255 ];
256
257 const dataProviderWithAuthors = {
258   getOne: (resource, params) =>
259     Promise.resolve({
260       data: {
261         id: 1,
262         title: 'War and Peace',
263         author: 1,
264         summary:
265           "War and Peace broadly focuses on Napoleon's invasion of
266           Russia, and the impact it had on Tsarist society. The book explores themes such as
267           revolution, revolution and empire, the growth and decline of various states and the
268           impact it had on their economies, culture, and society.",
269         year: 1869,
270       },
271     }),
272   getMany: (resource, params) =>

```

```

270     Promise.resolve({
271       data: authors.filter(author => params.ids.includes(author.id)),
272     }),
273   getList: (resource, params) =>
274     new Promise(resolve => {
275
276       if (params.filter.q == undefined) {
277         setTimeout(
278           () =>
279             resolve({
280               data: authors,
281               total: authors.length,
282             }),
283           500
284         );
285         return;
286       }
287
288       const filteredAuthors = authors.filter(author =>
289         author.name
290           .toLowerCase()
291           .includes(params.filter.q.toLowerCase())
292       );
293
294       setTimeout(
295         () =>
296           resolve({
297             data: filteredAuthors,
298             total: filteredAuthors.length,
299           }),
300         500
301       );
302     }),
303   update: (resource, params) => Promise.resolve(params),
304   create: (resource, params) => {
305     const newAuthor = {
306       id: authors.length + 1,
307       name: params.data.name,
308       language: params.data.language,
309     };
310     authors.push(newAuthor);
311     return Promise.resolve({ data: newAuthor });
312   },
313 } as any;
314
315 const BookEditWithReference = () => (
316   <Edit
317     mutationMode="pessimistic"
318     mutationOptions={{
319       onSuccess: data => {
320         console.log(data);
321       },
322     }}
323   >
324     <SimpleForm>
325       <ReferenceInput reference="authors" source="author">
326         <AutocompleteInput fullWidth />
327       </ReferenceInput>
328     </SimpleForm>
329   </Edit>
330 );
331
332 export const InsideReferenceInput = () => (
333   <Admin dataProvider={dataProviderWithAuthors} history={history}>
334     <Resource name="authors" />
335     <Resource name="books" edit={BookEditWithReference} />
336   </Admin>
337 );

```

```

338
339 const CreateAuthor = () => {
340     const { filter, onCancel, onCreate } = useCreateSuggestionContext();
341     const [name, setName] = React.useState(filter || '');
342     const [language, setLanguage] = React.useState('');
343     const [create] = useCreate();
344
345     const handleSubmit = event => {
346         event.preventDefault();
347         create(
348             'authors',
349             {
350                 data: {
351                     name,
352                     language,
353                 },
354             },
355             {
356                 onSuccess: ({ data }) => {
357                     setName('');
358                     setLanguage('');
359                     onCreate(data);
360                 },
361             }
362         );
363     };
364
365     return (
366         <Dialog open onClose={onCancel}>
367             <form onSubmit={handleSubmit}>
368                 <DialogContent>
369                     <Stack gap={4}>
370                         <TextField
371                             name="name"
372                             label="The author name"
373                             value={name}
374                             onChange={event => setName(event.target.value)}
375                             autoFocus
376                         />
377                         <TextField
378                             name="language"
379                             label="The author language"
380                             value={language}
381                             onChange={event => setLanguage(event.target.value)}
382                             autoFocus
383                         />
384                     </Stack>
385                 </DialogContent>
386                 <DialogActions>
387                     <Button type="submit">Save</Button>
388                     <Button onClick={onCancel}>Cancel</Button>
389                 </DialogActions>
390             </form>
391         </Dialog>
392     );
393 };
394
395 const BookEditWithReferenceAndCreationSupport = () => (
396     <Edit
397         mutationMode="pessimistic"
398         mutationOptions={{
399             onSuccess: data => {
400                 console.log(data);
401             },
402         }}
403     >
404         <SimpleForm>
405             <ReferenceInput reference="authors" source="author">
406                 <AutocompleteInput create={CreateAuthor} fullWidth />

```

```
407         </ReferenceInput>
408     </SimpleForm>
409 </Edit>
410 );
411
412 export const InsideReferenceInputWithCreationSupport = () => (
413     <Admin dataProvider={dataProviderWithAuthors} history={history}>
414         <Resource name="authors" />
415         <Resource name="books" edit={BookEditWithReferenceAndCreationSupport} />
416     </Admin>
417 );
```