

Country.js

```
import React, {Suspense, lazy} from 'react'

const SoftBox = lazy(() => import("components/SoftBox"));
const SoftTypography = lazy(() => import("components/SoftTypography"));
const SoftSelect = lazy(() => import("components/SoftSelect"));
import {Country, State, City} from 'country-state-city';
const Countrycomponent = (Props) => {
  console.log("formDataProps..", Props);
  const {formDataProps} = Props;
  console.log("formDataProps....", formDataProps);
  const { formField, values, errors, touched, setFieldValue } = formDataProps;
  const { address1, address2, city, state, zip} = formField;
  // const Country = require('country-state-city').Country;
  // const countries = Country?.getAllCountries();
  const countries = Country?.getAllCountries();
  console.log("countries..", countries);
  const updatedCountries = countries?.map((country) => ({
    label: country.name,
    value: country.id,
    ...country,
  }));
  return (
    <Suspense>
    <SoftBox>
    <SoftBox mb={1} ml={0.5} lineHeight={0} display="inline-block">
    <SoftTypography
      component="label"
      variant="caption"
      fontWeight="bold"
      textTransform="capitalize"
    >
    Country
    </SoftTypography>
    </SoftBox>
    <SoftSelect
      menuPortalTarget={document.body}
      menuPosition={"fixed"}
      placeholder="Select your Country"
      options={updatedCountries}
      onChange={(value) => {
        setFieldValue("country", value);
      }}
    </SoftSelect>
    </SoftBox>
    </Suspense>
  )
}
```

```
export default Countrycomponent;
```

State.js

```
import React, {Suspense, lazy} from 'react'
const SoftBox = lazy(() => import("components/SoftBox"));
const SoftTypography = lazy(() => import("components/SoftTypography"));
const SoftSelect = lazy(() => import("components/SoftSelect"));
import {Country, State, City} from 'country-state-city';

const Statecomponent = (Props) => {
  const {formDataProps} = Props;
  console.log("formDataProps...", formDataProps);
  const { formField, values, errors, touched, setFieldValue } = formDataProps;
  const { address1, address2, city, state, zip } = formField;
  const updatedStates = (countryId) =>
  State?.getStatesOfCountry(countryId).map((state) => ({
    label: state.name,
    value: state.id,
    ...state,
  }));
  return (
    <Suspense>
    <SoftBox>
    <SoftBox mb={1} ml={0.5} lineHeight={0} display="inline-block">
    <SoftTypography
      component="label"
      variant="caption"
      fontWeight="bold"
      textTransform="capitalize"
    >
    State
    </SoftTypography>
    </SoftBox>
    <SoftSelect
      menuPortalTarget={document.body}
      menuPosition="fixed"
      type={state.type}
      placeholder={state.placeholder}
      options={updatedStates(values.country ? values.country.isoCode : null)}
      value={values.state}
      onChange={(value) => {
        setFieldValue("state", value);
      }}
    />
    </SoftBox>
    </Suspense>
  )
}
```

City.js

```
import React, {Suspense, lazy} from 'react'
const SoftBox = lazy(() => import("components/SoftBox"));
const SoftTypography = lazy(() => import("components/SoftTypography"));
const SoftSelect = lazy(() => import("components/SoftSelect"));
import {Country, State, City} from 'country-state-city';

function Citycomponent(Props) {
  const {formDataProps} = Props;
  console.log("formDataProps...", formDataProps);
  const { formField, values, errors, touched, setFieldValue } = formDataProps;
  const { address1, address2, city, state, zip } = formField;
  const updatedCities = (stateId) =>
    City?.getCitiesOfState(stateId?.countryCode, stateId?.isoCode).map((city) =>
      ({
        label: city.name,
        value: city.id,
        ...city,
      }));
  return (
    <Suspense>
      <SoftBox mb={1} ml={0.5} lineHeight={0} display="inline-block">
        <SoftTypography
          component="label"
          variant="caption"
          fontWeight="bold"
          textTransform="capitalize"
        >
          City
        </SoftTypography>
      </SoftBox>
      <SoftSelect
        menuPortalTarget={document.body}
        menuPosition={"fixed"}
        type={city.type}
        placeholder={city.placeholder}
        options={updatedCities(values.state ? values.state : null)}
        value={values.city}
        onChange={(value) => setFieldValue("city", value)}
      />
    </Suspense>
  )
}
```

```
export default Citycomponent;
```