

```

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import CategoryService from "services/CategoryService";

export const getByJobPositionAPI = createAsyncThunk(
  "category/getByJobPositionAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByJobPosition();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getByJobSpecializeAPI = createAsyncThunk(
  "category/getByJobSpecializeAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByJobSpecialize();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getByDepartmentAPI = createAsyncThunk(
  "category/getByDepartmentAPI",
  async (branchId, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByDepartment(branchId);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getByJobTitleAPI = createAsyncThunk(
  "category/getByJobTitleAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByJobTitle();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getByWorkStatusAPI = createAsyncThunk(
  "category/getByWorkStatusAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByWorkStatus();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getByHisDepartmentAPI = createAsyncThunk(
  "category/getByHisDepartmentAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByHisDepartment();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

    }
  };
};

export const getByDepartmentLevelAPI = createAsyncThunk(
  "category/getByDepartmentLevelAPI",
  async (data, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getByDepartmentLevel();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getDepartmentByBranch = createAsyncThunk(
  "category/getDepartmentByBranch",
  async (id, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getDepartmentByBranch(id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getSpecialityAPI = createAsyncThunk(
  "category/getSpecialityAPI",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getSpecialityAPI();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getTimeSlotAPI = createAsyncThunk(
  "category/getTimeSlotAPI",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getTimeSlotAPI();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getHospitalApi = createAsyncThunk(
  "category/getHospitalApi",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getHospitalApi();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getBagetApi = createAsyncThunk(
  "category/getBagetApi",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getBagetApi();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const getProvinceApi = createAsyncThunk(
  "category/getProvinceApi",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getProvinceApi();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getDistrictApi = createAsyncThunk(
  "category/getDistrictApi",
  async (provinceId, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getDistrictApi(provinceId);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getWardApi = createAsyncThunk(
  "category/getWardApi",
  async (districtId, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getWardApi(districtId);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getNationApi = createAsyncThunk(
  "category/getNationApi",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getNationApi();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllLoaiCalamViec = createAsyncThunk(
  "category/getAllLoaiCalamViec",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getAllLoaiCalamViec();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllSymbol = createAsyncThunk(
  "category/getAllSymbol",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getAllSymbol();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllReason = createAsyncThunk(
  "category/getAllReason",
  async (_, { rejectWithValue }) => {
    try {

```

```

    const response = await CategoryService.getAllReason();
    return response.data;
  } catch (err) {
    return rejectWithValue(err.message || "Error");
  }
}
);

export const getAllEmployeeCate = createAsyncThunk(
  "category/getAllEmployeeCate",
  async (_, { rejectWithValue }) => {
    try {
      const response = await CategoryService.getAllEmployee();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

const initialState = {
  loading: false,
  positionCateList: [],
  titleCateList: [],
  workStatusList: [],
  departmentList: [],
  hisDepartmentList: [],
  departmentLevelSlice: [],
  authRoleList: [],
  authPermitList: [],
  deptByBranchList: [],
  specializeCateList: [],
  timeSlotList: [],
  hospitalList: [],
  bagetList: [],
  provinceList: [],
  districtList: [],
  wardList: [],
  nationList: [],
  loaicalamviecList: [],
  symbolList: [],
  reasonList: [],
  employeeList: [],
};

export const departmentSlice = createSlice({
  name: "category",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(getByJobPositionAPI.pending, (state) => {
        state.loading = true;
      })
      .addCase(getByJobPositionAPI.fulfilled, (state, action) => {
        state.loading = false;
        state.positionCateList = action.payload;
      })
      .addCase(getByJobPositionAPI.rejected, (state, action) => {
        state.loading = false;
      })
      .addCase(getByJobSpecializeAPI.pending, (state) => {
        state.loading = true;
      })
      .addCase(getByJobSpecializeAPI.fulfilled, (state, action) => {
        state.loading = false;
        state.specializeCateList = action.payload;
      })
      .addCase(getByJobSpecializeAPI.rejected, (state, action) => {
        state.loading = false;
      });
  }
});

```

```

})
.addCase(getByJobTitleAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getByJobTitleAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.titleCatelist = action.payload;
})
.addCase(getByJobTitleAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getByDepartmentAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getByDepartmentAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.departmentList = action.payload;
})
.addCase(getByDepartmentAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getByWorkStatusAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getByWorkStatusAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.workStatusList = action.payload;
})
.addCase(getByWorkStatusAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getByHisDepartmentAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getByHisDepartmentAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.hisDepartmentList = action.payload;
})
.addCase(getByHisDepartmentAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getByDepartmentLevelAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getByDepartmentLevelAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.departmentLevelList = action.payload;
})
.addCase(getByDepartmentLevelAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getDepartmentByBranch.pending, (state) => {
  state.loading = true;
})
.addCase(getDepartmentByBranch.fulfilled, (state, action) => {
  state.loading = false;
  state.deptByBranchList = action.payload;
})
.addCase(getDepartmentByBranch.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getSpecialityAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getSpecialityAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.specializeCateList = action.payload;
})
.addCase(getSpecialityAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getHospitalApi.pending, (state) => {
  state.loading = true;
})

```

```

.addCase(getHospitalApi.fulfilled, (state, action) => {
  state.loading = false;
  state.hospitalList = action.payload;
})
.addCase(getHospitalApi.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getTimeSlotAPI.pending, (state) => {
  state.loading = true;
})
.addCase(getTimeSlotAPI.fulfilled, (state, action) => {
  state.loading = false;
  state.timeSlotList = action.payload;
})
.addCase(getTimeSlotAPI.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getBagetApi.pending, (state) => {
  state.loading = true;
})
.addCase(getBagetApi.fulfilled, (state, action) => {
  state.loading = false;
  state.bagetList = action.payload;
})
.addCase(getBagetApi.rejected, (state, action) => {
  state.loading = false;
})

.addCase(getProvinceApi.pending, (state) => {
  state.loading = true;
})
.addCase(getProvinceApi.fulfilled, (state, action) => {
  state.loading = false;
  state.provinceList = action.payload;
})
.addCase(getProvinceApi.rejected, (state, action) => {
  state.loading = false;
})

.addCase(getDistrictApi.pending, (state) => {
  state.loading = true;
})
.addCase(getDistrictApi.fulfilled, (state, action) => {
  state.loading = false;
  state.districtList = action.payload;
})
.addCase(getDistrictApi.rejected, (state, action) => {
  state.loading = false;
})

.addCase(getWardApi.pending, (state) => {
  state.loading = true;
})
.addCase(getWardApi.fulfilled, (state, action) => {
  state.loading = false;
  state.wardList = action.payload;
})
.addCase(getWardApi.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getNationApi.pending, (state) => {
  state.loading = true;
})
.addCase(getNationApi.fulfilled, (state, action) => {
  state.loading = false;
  state.nationList = action.payload.map((x) => ({
    value: x.id,
    label: x.name,
  }));
})
.addCase(getNationApi.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getAllLoaiCalamViec.pending, (state) => {

```

```

      state.loading = true;
    })
    .addCase(getAllLoaiCalamViec.fulfilled, (state, action) => {
      state.loading = false;
      state.loaicalamviecList = action.payload;
    })
    .addCase(getAllLoaiCalamViec.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllSymbol.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllSymbol.fulfilled, (state, action) => {
      state.loading = false;
      state.symbolList = action.payload;
    })
    .addCase(getAllSymbol.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllReason.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllReason.fulfilled, (state, action) => {
      state.loading = false;
      state.reasonList = action.payload;
    })
    .addCase(getAllReason.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllEmployeeCate.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllEmployeeCate.fulfilled, (state, action) => {
      state.loading = false;
      state.employeeList = action.payload;
    })
    .addCase(getAllEmployeeCate.rejected, (state, action) => {
      state.loading = false;
    });
  },
});

```

```

export const { showLoading } = departmentSlice.actions;

```

```

export default departmentSlice.reducer;

```