

```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import dinhKhoanService from "services/DinhKhoan/DinhKhoanService";
import { cloneDeep } from "lodash";
```

```
export const getDinhKhoan = createAsyncThunk(
  "dinhKhoan/getDinhKhoan",
  async (data, { rejectWithValue }) => {
    try {
      const response = await dinhKhoanService.Get(data);
      return response.data;
    } catch (err) {
      console.error("API call failed:", err);
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const getCateDinhKhoan = createAsyncThunk(
  "dinhKhoan/getCateDinhKhoan",
  async (_, { rejectWithValue }) => {
    try {
      const response = await dinhKhoanService.GetCate();
      return response.data;
    } catch (err) {
      console.error("API call failed:", err);
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const updateDinhKhoan = createAsyncThunk(
  "dinhKhoan/updateDinhKhoan",
  async (data, { rejectWithValue, dispatch }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await dinhKhoanService.Update(data);
      if (onSuccess) onSuccess(response);

      return response.data;
    } catch (error) {
      console.error("API call failed:", error);
      return rejectWithValue(error.message || "Error");
    }
  }
);
```

```
export const createDinhKhoan = createAsyncThunk(
  "dinhKhoan/createDinhKhoan",
  async (data, { rejectWithValue, dispatch }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await dinhKhoanService.Create(data);
      if (onSuccess) onSuccess(response.data);

      return response.data;
    } catch (error) {
      console.error("API call failed:", error);
      return rejectWithValue(error.description || "Error");
    }
  }
);
```

```
export const deleteDinhKhoan = createAsyncThunk(
  "dinhKhoan/deleteDinhKhoan",
  async (data, { rejectWithValue }) => {
    const { onSuccess, id } = data;
    try {
```

```

    const response = await dinhKhoanService.Delete(id);
    if (onSuccess) onSuccess(response);
    return id;
  } catch (error) {
    console.error("API call failed:", error);
    return rejectWithValue(error.message || "Error");
  }
}
);
export const getDinhKhoanById = createAsyncThunk(
  "dinhKhoan/getDinhKhoanById",
  async (data, { rejectWithValue }) => {
    const { onSuccess, id } = data;
    try {
      const response = await dinhKhoanService.GetById(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (error) {
      console.error("API call failed:", error);
      return rejectWithValue(error.message || "Error");
    }
  }
);

const initialState = {
  loading: false,
  dinhKhoanList: [],
  dinhKhoanCate: [],
  dinhKhoanDetail: {
    loading: false,
    data: null,
  },
};

const dinhKhoanSlice = createSlice({
  name: "dinhKhoan",
  initialState,
  reducers: {},
  extraReducers: (builder) => {
    builder
      .addCase(getCateDinhKhoan.pending, (state) => {
        state.loading = true;
      })
      .addCase(getCateDinhKhoan.fulfilled, (state, action) => {
        state.loading = false;
        state.dinhKhoanCate = action.payload;
      })
      .addCase(getCateDinhKhoan.rejected, (state) => {
        state.loading = false;
      })
      .addCase(getDinhKhoan.pending, (state) => {
        state.loading = true;
      })
      .addCase(getDinhKhoan.fulfilled, (state, action) => {
        state.loading = false;
        state.dinhKhoanList = action.payload;
      })
      .addCase(getDinhKhoan.rejected, (state) => {
        state.loading = false;
      })
      .addCase(updateDinhKhoan.pending, (state) => {
        state.loading = true;
      })
      .addCase(updateDinhKhoan.fulfilled, (state) => {
        state.loading = false;
      })
      .addCase(updateDinhKhoan.rejected, (state) => {
        state.loading = false;
      })
      .addCase(createDinhKhoan.pending, (state) => {
        state.loading = true;
      })
      .addCase(createDinhKhoan.fulfilled, (state) => {
        state.loading = false;
      })
  }
});

```

```

    })
    .addCase(createDinhKhoan.rejected, (state) => {
      state.loading = false;
    })
    .addCase(deleteDinhKhoan.pending, (state) => {
      state.loading = true;
    })
    .addCase(deleteDinhKhoan.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(deleteDinhKhoan.rejected, (state) => {
      state.loading = false;
    })

    .addCase(getDinhKhoanById.pending, (state) => {
      state.dinhKhoanDetail = { ...state.dinhKhoanDetail, loading: true };
    })
    .addCase(getDinhKhoanById.fulfilled, (state, action) => {
      state.dinhKhoanDetail = { data: action.payload, loading: false };
    })
    .addCase(getDinhKhoanById.rejected, (state) => {
      state.dinhKhoanDetail = { ...state.dinhKhoanDetail, loading: false };
    });
  },
});
export const {} = dinhKhoanSlice.actions;

export default dinhKhoanSlice.reducer;

```