

```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { notification } from "antd";
import { cloneDeep } from "lodash";
import ratingService from "services/RatingService.js";
```

```
export const getByCodeRatingApi = createAsyncThunk(
  "rating/getByCoderatingApi",
  async (code, { rejectWithValue }) => {
    try {
      const response = await ratingService.ratingGetByCodeAPI(code);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const addRatingApi = createAsyncThunk(
  "rating/addRatingApi",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ratingService.ratingSubmitAPI(data);
      notification.success({
        message: "Thông báo ",
        description: "Chúng tôi đã ghi nhận ý kiến của bạn."
      });
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const checkTheApi = createAsyncThunk(
  "esign/checkThe",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ratingService.checkTheAPI(data);
      // notification.success({
      //   message: "Thông báo ",
      //   description: "Chúng tôi đã ghi nhận ý kiến của bạn."
      // });
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      const { onReject } = data;
      console.log('err', err);
      let msg = err.message;
      if (err.response.status === 600) msg = err.response.data.description;
      if (onReject) onReject(msg);
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const esignSubmitApi = createAsyncThunk(
  "esign/submit",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ratingService.esignSubmit(data);
      // notification.success({
      //   message: "Thông báo ",
```

```

        // description: "Chúng tôi đã ghi nhận ý kiến của bạn."
        // });
        if (onSuccess) onSuccess(response);
        return response.data;
    } catch (err) {
        return rejectWithValue(err.message || "Error");
    }
}
);

const initialState = {
    loading: false,
    customerInfo: {},
};

export const RatingSlice = createSlice({
    name: "rating",
    initialState,
    reducers: {
        showLoading: (state) => {
            state.loading = true;
        },
    },
    extraReducers: (builder) => {
        builder
            .addCase(getByCodeRatingApi.pending, (state) => {
                state.loading = true;
            })
            .addCase(getByCodeRatingApi.fulfilled, (state, action) => {
                state.loading = false;
                state.customerInfo = action.payload;
            })
            .addCase(getByCodeRatingApi.rejected, (state, action) => {
                state.loading = false;
            })
            .addCase(addRatingApi.pending, (state) => {
                state.loading = true;
            })
            .addCase(addRatingApi.fulfilled, (state, action) => {
                state.loading = false;
            })
            .addCase(addRatingApi.rejected, (state, action) => {
                state.loading = false;
            });
    },
});

export const { showLoading } = RatingSlice.actions;

export default RatingSlice.reducer;

```