

```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { cloneDeep } from "lodash";
import DichVuTaiNhaService from "services/DichVuTaiNhaService";
import DoctorManagementService from "services/DoctorManagementService";
import DichVuTaiNhaCategoryService from "services/DichVuTaiNhaCategoryService";
```

```
export const getAllDichVuTaiNha = createAsyncThunk(
  "dichVuTaiNha/getAllDichVuTaiNha",
  async (data, { rejectWithValue }) => {
    try {
      const response = await DichVuTaiNhaService.searchGrid(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const getAllDichVuTaiNhaCategory = createAsyncThunk(
  "dichVuTaiNha/getAllDichVuTaiNhaCategory",
  async (data, { rejectWithValue }) => {
    try {
      const response =
        await DichVuTaiNhaCategoryService.getAllDichVuTaiNhaCategory();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const getDichVuTaiNhaById = createAsyncThunk(
  "dichVuTaiNha/getDichVuTaiNhaById",
  async (id, { rejectWithValue }) => {
    try {
      const response = await DichVuTaiNhaService.getDichVuTaiNhaById(id);
      return response.data.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const createDichVuTaiNha = createAsyncThunk(
  "dichVuTaiNha/createDichVuTaiNha",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DichVuTaiNhaService.create(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```
export const updateDichVuTaiNha = createAsyncThunk(
  "dichVuTaiNha/updateDichVuTaiNha",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DichVuTaiNhaService.update(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
```

```

    }
  }
);

export const deletedDichVuTaiNha = createAsyncThunk(
  "dichVuTaiNha/deletedDichVuTaiNhaById",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess, id } = data;
      const response = await DichVuTaiNhaService.delete(id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const changeStateDichVuTaiNha = createAsyncThunk(
  "dichVuTaiNha/changeStateDichVuTaiNha",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await DichVuTaiNhaService.changeStateDichVuTaiNha(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

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

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

const initialState = {
  loading: false,
  dichVuTaiNhaList: [],
  dichVuTaiNhaDetail: {},
  hospitalList: [],
  doctorList: [],
  dichVuTaiNhaCategoryList: [],
  codeDichVuTaiNha: null,
};

export const DichVuTaiNhaSlice = createSlice({
  name: "dichvutainha",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
    setHisInfo: (state, action) => {

```

```

state.hisInfoList = action.payload;
},
},
extraReducers: (builder) => {
  builder
    .addCase(getAllDichVuTaiNha.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllDichVuTaiNha.fulfilled, (state, action) => {
      state.loading = false;
      state.dichVuTaiNhaList = action.payload;
    })
    .addCase(getAllDichVuTaiNha.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllDichVuTaiNhaCategory.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllDichVuTaiNhaCategory.fulfilled, (state, action) => {
      state.loading = false;
      state.dichVuTaiNhaCategoryList = action.payload;
    })
    .addCase(getAllDichVuTaiNhaCategory.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getDichVuTaiNhaById.pending, (state) => {
      state.loading = true;
    })
    .addCase(getDichVuTaiNhaById.fulfilled, (state, action) => {
      state.loading = false;
      state.dichVuTaiNhaDetail = action.payload;
    })
    .addCase(getDichVuTaiNhaById.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(createDichVuTaiNha.pending, (state) => {
      state.loading = true;
    })
    .addCase(createDichVuTaiNha.fulfilled, (state, action) => {
      state.loading = false;
      state.codeDichVuTaiNha = action.payload;
    })
    .addCase(createDichVuTaiNha.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(updateDichVuTaiNha.pending, (state) => {
      state.loading = true;
    })
    .addCase(updateDichVuTaiNha.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(updateDichVuTaiNha.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(deletedDichVuTaiNha.pending, (state) => {
      state.loading = true;
    })
    .addCase(deletedDichVuTaiNha.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(deletedDichVuTaiNha.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(changeStateDichVuTaiNha.pending, (state) => {
      state.loading = true;
    })
    .addCase(changeStateDichVuTaiNha.fulfilled, (state, action) => {
      state.loading = false;
    })
    .addCase(changeStateDichVuTaiNha.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllHospital.pending, (state) => {
      state.loading = true;
    })

```

```

    })
    .addCase(getAllHospital.fulfilled, (state, action) => {
      state.loading = false;
      state.hospitalList = action.payload;
    })
    .addCase(getAllHospital.rejected, (state, action) => {
      state.loading = false;
    })
    .addCase(getAllDoctor.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllDoctor.fulfilled, (state, action) => {
      state.loading = false;
      state.doctorList = action.payload;
    })
    .addCase(getAllDoctor.rejected, (state, action) => {
      state.loading = false;
    });
  },
});

export const { showLoading, setHisInfo } = DichVuTaiNhaSlice.actions;

export default DichVuTaiNhaSlice.reducer;

```