D:\应用程序案例\spartacus\projects\core\src\state\utils\loader\loader.action.ts

export interface LoaderMeta {

entityType: string;

loader: {

load?: boolean;

error?: any;

success?: boolean;

};

}

export interface LoaderAction extends Action {

readonly payload?: any;

readonly meta?: LoaderMeta;

}

export class LoaderLoadAction implements LoaderAction {

type = LOADER\_LOAD\_ACTION;

readonly meta: LoaderMeta;

constructor(entityType: string) {

this.meta = loadMeta(entityType);

}

}

D:\应用程序案例\spartacus\projects\core\src\state\utils\entity\entity.action.ts

export interface EntityMeta {

entityType: string;

entityId: string | string[];

entityRemove?: boolean;

}

export interface EntityAction extends Action {

readonly payload?: any;

readonly meta?: EntityMeta;

}

export class EntityRemoveAction implements EntityAction {

type = ENTITY\_REMOVE\_ACTION;

readonly meta: EntityMeta;

constructor(entityType: string, id: string | string[]) {

this.meta = entityRemoveMeta(entityType, id);

}

}

D:\应用程序案例\spartacus\projects\core\src\state\utils\entity-loader\entity-loader.action.ts

export interface EntityLoaderMeta extends EntityMeta, LoaderMeta {}

export interface EntityLoaderAction extends Action {

readonly payload?: any;

readonly meta?: EntityLoaderMeta;

}

export class EntityLoadAction implements EntityLoaderAction {

type = ENTITY\_LOAD\_ACTION;

readonly meta: EntityLoaderMeta;

constructor(entityType: string, id: string | string[]) {

this.meta = entityLoadMeta(entityType, id);

}

}

D:\spartacus\projects\core\src\state\utils\processes-loader\processes-loader.action.ts

export interface ProcessesLoaderMeta extends LoaderMeta {

entityType: string;

processesCountDiff?: number;

}

export interface ProcessesLoaderAction extends Action {

readonly payload?: any;

readonly meta?: ProcessesLoaderMeta;

}

export class ProcessesLoaderResetAction implements ProcessesLoaderAction {

type = PROCESSES\_LOADER\_RESET\_ACTION;

readonly meta: ProcessesLoaderMeta;

constructor(entityType: string) {

this.meta = processesLoaderResetMeta(entityType);

}

}

D:\spartacus\projects\core\src\state\utils\entity-processes-loader\entity-processes-loader.action.ts

export interface EntityProcessesLoaderMeta extends EntityMeta, ProcessesLoaderMeta {}

export interface EntityProcessesLoaderAction extends Action {

readonly payload?: any;

readonly meta?: EntityProcessesLoaderMeta;

}

export class EntityProcessesLoaderResetAction

implements EntityProcessesLoaderAction {

type = ENTITY\_PROCESSES\_LOADER\_RESET\_ACTION;

readonly meta: EntityProcessesLoaderMeta;

constructor(entityType: string, id: string | string[]) {

this.meta = entityProcessesLoaderResetMeta(entityType, id);

}

}

D:\spartacus\projects\core\src\state\utils\scoped-loader\entity-scoped-loader.actions.ts

export interface EntityScopedLoaderMeta extends EntityLoaderMeta {

scope?: string;

}

export interface EntityScopedLoaderAction extends Action {

readonly payload?: any;

readonly meta?: EntityScopedLoaderMeta;

}

export class EntityScopedLoadAction implements EntityScopedLoaderAction {

type = ENTITY\_LOAD\_ACTION;

readonly meta: EntityScopedLoaderMeta;

constructor(entityType: string, id: string | string[], scope?: string) {

this.meta = entityScopedLoadMeta(entityType, id, scope);

}

}

