

docs\README.md

frontend

This is a [Next.js](#) project bootstrapped with [create-next-app](#).

Getting Started

First, run the development server:

```
npm run dev
# or
yarn dev
# or
pnpm dev
# or
bun dev
```

Open <http://localhost:3000> with your browser to see the result.

You can start editing the page by modifying `app/page.tsx`. The page auto-updates as you edit the file.

This project uses [next/font](#) to automatically optimize and load [Geist](#), a new font family for Vercel.

Learn More

To learn more about Next.js, take a look at the following resources:

- [Next.js Documentation](#) - learn about Next.js features and API.
- [Learn Next.js](#) - an interactive Next.js tutorial.

You can check out [the Next.js GitHub repository](#) - your feedback and contributions are welcome!

Deploy on Vercel

The easiest way to deploy your Next.js app is to use the [Vercel Platform](#) from the creators of Next.js.

Check out our [Next.js deployment documentation](#) for more details.

docs\components\AutoLoginHandler\README.md

frontend

[frontend](#) / components/AutoLoginHandler

components/AutoLoginHandler

Functions

- [handleAutoLogin](#)
- [handleAutoLoginWithRerouteToLoginPage](#)

docs\components\AutoLoginHandler\functions\handleAutoLogin.md

frontend

[frontend](#) / [components/AutoLoginHandler](#) / handleAutoLogin

Function: handleAutoLogin()

```
handleAutoLogin(): Promise < void >
```

Defined in: [components/AutoLoginHandler.tsx:11](#)

Automatically logs in a user by retrieving the access token from the cookies. If the access token is invalid, logs out the user.

Returns

```
Promise < void >
```

A promise resolved when the login process is finished.

docs\components\AutoLoginHandler\functions\handleAutoLoginWithRerouteToLc

frontend

Function: handleAutoLoginWithRerouteToLoginPage()

`handleAutoLoginWithRerouteToLoginPage(): Promise < void >`

Defined in: [components/AutoLoginHandler.tsx:29](#)

Automatically logs in a user by retrieving the access token from the cookies. If the access token is invalid, logs out the user and redirects to the login page.

Returns

`Promise < void >`

A promise resolved when the login process is finished.

docs\components\Avatar\README.md

[frontend](#)

components/Avatar

Functions

- [Avatar](#)

docs\components\Avatar\functions\Avatar.md

[frontend](#)

Function: Avatar()

`Avatar(size): Element`

Defined in: [components/Avatar.tsx:12](#)

Returns a div element with a background color derived from the name and surname, and displays the name and surname inside the div.

Parameters

size

size of the div

name?

`string = ""`

size?

`number = 32`

surname?

`string = ""`

Returns

`Element`

docs\components\CategoryItem\README.md

[frontend](#)

components/CategoryItem

Functions

- [CategoryItem](#)

docs\components\CategoryItem\functions\CategoryItem.md

[frontend](#)

[frontend](#) / [components/CategoryItem](#) / CategoryItem

Function: CategoryItem()

```
CategoryItem( __namedParameters ): Element
```

Defined in: [components/CategoryItem.tsx:1](#)

Parameters

__namedParameters

id_category

number

label

string

onClick

any

selectable?

boolean = true

selected?

boolean = false

selectedItems?

number | null = null

selectedItemsOverride?

boolean | null = null

Returns

Element

docs\hooks\useCountdown\README.md

[frontend](#)

[frontend](#) / [hooks/useCountdown](#)

hooks/useCountdown

Functions

- [useCountdown](#)

docs\hooks\useCountdown\functions\useCountdown.md

[frontend](#)

[frontend](#) / [hooks/useCountdown](#) / useCountdown

Function: useCountdown()

```
useCountdown( iso ): string
```

Defined in: [hooks/useCountdown.tsx:5](#)

Parameters

iso

string

Returns

string

docs\modules.md

[frontend](#)

frontend

Modules

- [app/\(app\)/\(auth\)/logowanie/page](#)
- [app/\(app\)/\(auth\)/rejestracja/page](#)
- [app/\(app\)/\(authed\)/moje-aukcje/page](#)
- [app/\(app\)/\(authed\)/nowa-aukcja/page](#)
- [app/\(app\)/\(authed\)/profil/page](#)
- [app/\(app\)/\(authed\)/wygrana-aukcja/auktion-slug/page](#)
- [app/\(app\)/aukcja/auktion-slug/page](#)
- [app/\(app\)/layout](#)
- [app/\(app\)/page](#)
- [app/layout](#)
- [components/AuctionCard](#)
- [components/AuthTokenHandler](#)
- [components/Avatar](#)
- [components/CategoryItem](#)
- [components/icon/Add](#)
- [components/icon/Auctions](#)
- [components/icon/Icon](#)
- [components/icon/Logo](#)
- [components/icon/Ring](#)
- [components/icon/RingInfo](#)
- [components/icon/Search](#)
- [components/icon/template](#)
- [components/icon/Trash](#)
- [components/icon/User](#)
- [components/inputs/Button](#)
- [components/inputs/TextInput](#)
- [components/structure/Footer](#)
- [components/structure/Header](#)
- [hooks/useCountdown](#)
- [pages/AuctionPage](#)
- [pages/auth/LoginPage](#)
- [pages/auth/RegisterPage](#)
- [pages/HomePage](#)
- [pages/MyAuctionsPage](#)
- [pages/NewAuctionPage](#)
- [pages/ProfilePage](#)
- [pages/WonAuctionPage](#)
- [socket](#)
- [store/slices/auktionSlice](#)
- [store/slices/authSelector](#)
- [store/slices/authSlice.ts](#)
- [store/slices/categoriesSlice](#)
- [store/slices/userAuctionSlice](#)
- [store/store](#)
- [store/thunks/AuctionsThunk](#)
- [store/thunks/auth/AuthTokenThunk](#)
- [store/thunks/auth/ChangePasswordThunk](#)
- [store/thunks/auth/LoginThunk](#)
- [store/thunks/auth/RegisterThunk](#)
- [utils/formatCountdownPL](#)

docs\pages\AuctionPage\README.md

[frontend](#)

[frontend](#) / [pages/AuctionPage](#)

pages/AuctionPage

Functions

- [default](#)

docs\pages\AuctionPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/AuctionPage](#) / default

Function: default()

`default()`: *Element*

Defined in: [pages/AuctionPage.tsx:22](#)

Returns

Element

docs\pages\HomePage\README.md

[frontend](#)

[frontend](#) / [pages/HomePage](#)

pages/HomePage

Functions

- [default](#)

docs\pages\HomePage\functions\default.md

[frontend](#)

[frontend](#) / [pages/HomePage](#) / default

Function: default()

`default()`: *Element*

Defined in: [pages/HomePage.tsx:21](#)

Returns

Element

docs\pages\MyAuctionsPage\README.md

[frontend](#)

[frontend](#) / [pages/MyAuctionsPage](#)

pages/MyAuctionsPage

Functions

- [default](#)

docs\pages\MyAuctionsPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/MyAuctionsPage](#) / default

Function: default()

`default()`: *Element*

Defined in: [pages/MyAuctionsPage.tsx:134](#)

Returns

Element

docs\pages\NewAuctionPage\README.md

[frontend](#)

[frontend](#) / [pages/NewAuctionPage](#)

pages/NewAuctionPage

Functions

- [default](#)

docs\pages\NewAuctionPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/NewAuctionPage](#) / default

Function: default()

default(): Element

Defined in: [pages/NewAuctionPage.tsx:44](#)

Returns

Element

docs\pages\ProfilePage\README.md

[frontend](#)

[frontend](#) / [pages/ProfilePage](#)

pages/ProfilePage

Functions

- [default](#)

docs\pages\ProfilePage\functions\default.md

[frontend](#)

[frontend](#) / [pages/ProfilePage](#) / default

Function: default()

default(): Element

Defined in: [pages/ProfilePage.tsx:18](#)

Returns

Element

docs\pages\WonAuctionPage\README.md

[frontend](#)

[frontend](#) / [pages/WonAuctionPage](#)

pages/WonAuctionPage

Functions

- [default](#)

docs\pages\WonAuctionPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/WonAuctionPage](#) / default

Function: default()

`default()`: `Element`

Defined in: [pages/WonAuctionPage.tsx:17](#)

Returns

`Element`

docs\pages\auth\LoginPage\README.md

[frontend](#)

[frontend](#) / [pages/auth/LoginPage](#)

pages/auth/LoginPage

Functions

- [default](#)

docs\pages\auth\LoginPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/auth/LoginPage](#) / default

Function: default()

`default()`: `Element`

Defined in: [pages/auth/LoginPage.tsx:12](#)

Returns

`Element`

docs\pages\auth\RegisterPage\README.md

[frontend](#)

[frontend](#) / [pages/auth/RegisterPage](#)

pages/auth/RegisterPage

Functions

- [default](#)

docs\pages\auth\RegisterPage\functions\default.md

[frontend](#)

[frontend](#) / [pages/auth/RegisterPage](#) / default

Function: default()

`default(): Element`

Defined in: [pages/auth/RegisterPage.tsx:9](#)

Returns

Element

docs\socket\README.md

[frontend](#)

[frontend](#) / [socket](#)

socket

Variables

- [socket](#)

docs\socket\variables\socket.md

[frontend](#)

[frontend](#) / [socket](#) / [socket](#)

Variable: socket

`const socket: Socket < DefaultEventsMap , DefaultEventsMap >`

Defined in: [socket.ts:6](#)

docs\store\slices\auctionSlice\README.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#)

store/slices/auctionSlice

Type Aliases

- [Auction](#)

Variables

- [auctionsSlice](#)
- [selectFilteredAuctionsBySearch](#)
- [selectUnsoldAuctions](#)
- [selectUnsoldFilteredAuctionsBySearch](#)
- [selectUnsoldFilteredAuctionsBySearchAndCategory](#)
- [setAuctions](#)
- [setSearch](#)

Functions

- [selectAllAuctions](#)
- [selectSearch](#)

docs\store\slices\auctionSlice\functions\selectAllAuctions.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / [selectAllAuctions](#)

Function: selectAllAuctions()

`selectAllAuctions(state): Auction []`

Defined in: [store/slices/auctionSlice.ts:56](#)

Parameters

state

auctions

AuctionsState

Returns

[Auction](#) []

docs\store\slices\auctionSlice\functions\selectSearch.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / selectSearch

Function: selectSearch()

`selectSearch(state): string`

Defined in: [store/slices/auctionSlice.ts:62](#)

Parameters

state

auctions

AuctionsState

Returns

string

docs\store\slices\auctionSlice\type-aliases\Auction.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / Auction

Type Alias: Auction

`Auction = object`

Defined in: [store/slices/auctionSlice.ts:4](#)

Properties

categories

`categories: string []`

Defined in: [store/slices/auctionSlice.ts:5](#)

current_price

`current_price: number`

Defined in: [store/slices/auctionSlice.ts:6](#)

description

`description: string`

Defined in: [store/slices/auctionSlice.ts:7](#)

end_date

`end_date: string`

Defined in: [store/slices/auctionSlice.ts:8](#)

id_auction

`id_auction`: *number*

Defined in: [store/slices/auctionSlice.ts:9](#)

id_seller

`id_seller`: *number*

Defined in: [store/slices/auctionSlice.ts:10](#)

id_winner

`id_winner`: *number* | *null*

Defined in: [store/slices/auctionSlice.ts:11](#)

main_photo

`main_photo`: *string* | *null*

Defined in: [store/slices/auctionSlice.ts:12](#)

overtime

`overtime`: *number*

Defined in: [store/slices/auctionSlice.ts:13](#)

start_date

`start_date`: *string*

Defined in: [store/slices/auctionSlice.ts:14](#)

start_price

`start_price`: *number*

Defined in: [store/slices/auctionSlice.ts:15](#)

status

`status`: *string*

Defined in: [store/slices/auctionSlice.ts:16](#)

title

`title`: *string*

Defined in: [store/slices/auctionSlice.ts:17](#)

docs\store\slices\auctionSlice\variables\auctionsSlice.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / auctionsSlice

Variable: auctionsSlice

```
const auctionsSlice: Slice < AuctionsState, { setAuctions: void; setSearch: void; }, "auctions", "auctions",  
  SliceSelectors < AuctionsState >>
```

Defined in: [store/slices/auctionSlice.ts:30](#)

docs\store\slices\auctionSlice\variables\selectFilteredAuctionsBySearch.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / selectFilteredAuctionsBySearch

Variable: selectFilteredAuctionsBySearch

```
const selectFilteredAuctionsBySearch: ( state , ... params ) => Auction [] & object & object & object
```

Defined in: [store/slices/auctionSlice.ts:64](#)

docs\store\slices\auctionSlice\variables\selectUnsoldAuctions.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / selectUnsoldAuctions

Variable: selectUnsoldAuctions

```
const selectUnsoldAuctions: ( state , ... params ) => Auction [] & object & object & object
```

Defined in: [store/slices/auctionSlice.ts:58](#)

docs\store\slices\auctionSlice\variables\selectUnsoldFilteredAuctionsBySearch.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / selectUnsoldFilteredAuctionsBySearch

Variable: selectUnsoldFilteredAuctionsBySearch

```
const selectUnsoldFilteredAuctionsBySearch: ( state , ... params ) => Auction [] & object & object & object
```

Defined in: [store/slices/auctionSlice.ts:73](#)

docs\store\slices\auctionSlice\variables\selectUnsoldFilteredAuctionsBySearchAnd

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / selectUnsoldFilteredAuctionsBySearchAndCategory

Variable: selectUnsoldFilteredAuctionsBySearchAndCategory

```
const selectUnsoldFilteredAuctionsBySearchAndCategory: ( state , ... params ) => Auction [] & object & object & object
```

Defined in: [store/slices/auctionSlice.ts:82](#)

docs\store\slices\auctionSlice\variables\setAuctions.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / setAuctions

Variable: setAuctions

```
setAuctions: ActionCreatorWithPayload < Auction [], "auctions/setAuctions" >
```

Defined in: [store/slices/auctionSlice.ts:53](#)

docs\store\slices\auctionSlice\variables\setSearch.md

[frontend](#)

[frontend](#) / [store/slices/auctionSlice](#) / setSearch

Variable: setSearch

```
setSearch: ActionCreatorWithPayload < string , "auctions/setSearch" >
```

Defined in: [store/slices/auctionSlice.ts:53](#)

docs\store\slices\authSelector\README.md

[frontend](#)

[frontend](#) / store/slices/authSelector

store/slices/authSelector

Functions

- [formatAuthDate](#)
- [selectAuth](#)

docs\store\slices\authSelector\functions\formatAuthDate.md

[frontend](#)

[frontend](#) / [store/slices/authSelector](#) / formatAuthDate

Function: formatAuthDate()

```
formatAuthDate( iso ): string | null
```

Defined in: [store/slices/authSelector.ts:10](#)

Format an ISO date string to dd.mm.yyyy

Parameters

iso

ISO date string

```
string | null
```

Returns

```
string | null
```

- Formatted date string or null if iso is invalid

docs\store\slices\authSelector\functions\selectAuth.md

[frontend](#)

[frontend](#) / [store/slices/authSelector](#) / selectAuth

Function: selectAuth()

```
selectAuth( state ): object
```

Defined in: [store/slices/authSelector.ts:3](#)

Parameters

state

actions

```
AuctionsState = auctionsReducer
```

auth

```
{ access_token : string | null ; create_account_date : string | null ; email : string | null ; first_name : string | null ; isAuthenticated : boolean ; last_name : string | null ; phone_number : string | null ; } = authReducer
```

auth.access_token

```
string | null
```

auth.create_account_date

```
string | null
```

auth.email

```
string | null
```

auth.first_name

```
string | null
```

auth.isAuthenticated

```
boolean
```

auth.last_name

string | null

auth.phone_number

string | null

categories

CategoriesState = categoriesReducer

userAuctions

MyAuctionsState = userAuctionsReducer

Returns

object

access_token

access_token: string | null

create_account_date

create_account_date: string | null

email

email: string | null

first_name

first_name: string | null

isAuthenticated

isAuthenticated: boolean

last_name

last_name: string | null

phone_number

phone_number: string | null

docs\store\slices\authSlice.ts\README.md

[frontend](#)

[frontend](#) / store/slices/authSlice.ts

store/slices/authSlice.ts

Variables

- [authSlice](#)
- [login](#)
- [logout](#)

docs\store\slices\authSlice.ts\variables\authSlice.md

[frontend](#)

[frontend](#) / [store/slices/authSlice.ts](#) / authSlice

Variable: authSlice

```
const authSlice: Slice <{ access_token: string | null; create_account_date: string | null; email: string | null; first_name: string | null; isAuthenticated: boolean; last_name: string | null; phone_number: string | null; }, { Login: (state, action) => void; Logout: (state) => void; }, "auth", "auth", SliceSelectors <{ access_token: string | null; create_account_date: string | null; email: string | null; first_name: string | null; isAuthenticated: boolean; last_name: string | null; phone_number: string | null; }>>
```

Defined in: [store/slices/authSlice.ts:24](#)

docs\store\slices\authSlice.ts\variables\login.md

[frontend](#)

[frontend](#) / [store/slices/authSlice.ts](#) / login

Variable: login

login: *ActionCreatorWithPayload* < any , "auth/Login" >

Defined in: [store/slices/authSlice.ts:63](#)

docs\store\slices\authSlice.ts\variables\logout.md

[frontend](#)

[frontend](#) / [store/slices/authSlice.ts](#) / logout

Variable: logout

logout: *ActionCreatorWithoutPayload* < "auth/Logout" >

Defined in: [store/slices/authSlice.ts:63](#)

docs\store\slices\categoriesSlice\README.md

[frontend](#)

[frontend](#) / store/slices/categoriesSlice

store/slices/categoriesSlice

Type Aliases

- [Category](#)

Variables

- [categoriesSlice](#)
- [setCategories](#)
- [setSelectedCategoryId](#)

Functions

- [selectCategories](#)
- [selectSelectedCategory](#)
- [selectSelectedCategoryId](#)

docs\store\slices\categoriesSlice\functions\selectCategories.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / selectCategories

Function: selectCategories()

selectCategories(state): any

Defined in: [store/slices/categoriesSlice.ts:43](#)

Parameters

state

any

Returns

any

docs\store\slices\categoriesSlice\functions\selectSelectedCategory.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / selectSelectedCategory

Function: selectSelectedCategory()

`selectSelectedCategory(state): any`

Defined in: [store/slices/categoriesSlice.ts:45](#)

Parameters

state

any

Returns

any

docs\store\slices\categoriesSlice\functions\selectSelectedCategoryId.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / selectSelectedCategoryId

Function: selectSelectedCategoryId()

`selectSelectedCategoryId(state): any`

Defined in: [store/slices/categoriesSlice.ts:44](#)

Parameters

state

any

Returns

any

docs\store\slices\categoriesSlice\type-aliases\Category.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / Category

Type Alias: Category

`Category = object`

Defined in: [store/slices/categoriesSlice.ts:3](#)

Properties

category_name

`category_name: string`

Defined in: [store/slices/categoriesSlice.ts:5](#)

id_category

`id_category: number`

Defined in: [store/slices/categoriesSlice.ts:4](#)

docs\store\slices\categoriesSlice\variables\categoriesSlice.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / categoriesSlice

Variable: categoriesSlice

```
const categoriesSlice: Slice < CategoriesState, { setCategories: void; setSelectedCategoryId: void; }, "categories", "categories",  
  SliceSelectors < CategoriesState >>
```

Defined in: [store/slices/categoriesSlice.ts:18](#)

docs\store\slices\categoriesSlice\variables\setCategories.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / setCategories

Variable: setCategories

```
setCategories: ActionCreatorWithPayload < Category [], "categories/setCategories" >
```

Defined in: [store/slices/categoriesSlice.ts:40](#)

docs\store\slices\categoriesSlice\variables\setSelectedCategoryId.md

[frontend](#)

[frontend](#) / [store/slices/categoriesSlice](#) / setSelectedCategoryId

Variable: setSelectedCategoryId

```
setSelectedCategoryId: ActionCreatorWithPayload < number | null, "categories/setSelectedCategoryId" >
```

Defined in: [store/slices/categoriesSlice.ts:40](#)

docs\store\slices\userAuctionSlice\README.md

[frontend](#)

[frontend](#) / store/slices/userAuctionSlice

store/slices/userAuctionSlice

Variables

- [setArchivedAuctions](#)
- [setOwnAuctions](#)
- [setParticipatingAuctions](#)

Functions

- [selectArchivedAuctions](#)
- [selectOwnAuctions](#)
- [selectParticipatingAuctions](#)

docs\store\slices\userAuctionSlice\functions\selectArchivedAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / selectArchivedAuctions

Function: selectArchivedAuctions()

```
selectArchivedAuctions( state ): any
```

Defined in: [store/slices/userAuctionSlice.ts:54](#)

Parameters

state

any

Returns

any

docs\store\slices\userAuctionSlice\functions\selectOwnAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / selectOwnAuctions

Function: selectOwnAuctions()

`selectOwnAuctions(state): any`

Defined in: [store/slices/userAuctionSlice.ts:52](#)

Parameters

state

any

Returns

any

docs\store\slices\userAuctionSlice\functions\selectParticipatingAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / selectParticipatingAuctions

Function: selectParticipatingAuctions()

`selectParticipatingAuctions(state): any`

Defined in: [store/slices/userAuctionSlice.ts:53](#)

Parameters

state

any

Returns

any

docs\store\slices\userAuctionSlice\variables\setArchivedAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / setArchivedAuctions

Variable: setArchivedAuctions

`setArchivedAuctions: ActionCreatorWithPayload < Auction [], "myAuctions/setArchivedAuctions" >`

Defined in: [store/slices/userAuctionSlice.ts:47](#)

docs\store\slices\userAuctionSlice\variables\setOwnAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / setOwnAuctions

Variable: setOwnAuctions

`setOwnAuctions: ActionCreatorWithPayload < Auction [], "myAuctions/setOwnAuctions" >`

Defined in: [store/slices/userAuctionSlice.ts:47](#)

docs\store\slices\userAuctionSlice\variables\setParticipatingAuctions.md

[frontend](#)

[frontend](#) / [store/slices/userAuctionSlice](#) / setParticipatingAuctions

Variable: setParticipatingAuctions

setParticipatingAuctions: *ActionCreatorWithPayload* < [Auction](#) [], "myAuctions/setParticipatingAuctions" >

Defined in: [store/slices/userAuctionSlice.ts:47](#)

docs\store\store\README.md

[frontend](#)

[frontend](#) / store/store

store/store

Type Aliases

- [AppDispatch](#)
- [AppStore](#)
- [RootState](#)

Variables

- [store](#)
- [useAppSelector](#)

Functions

- [useAppDispatch](#)

References

default

Renames and re-exports [store](#)

docs\store\store\functions\useAppDispatch.md

[frontend](#)

[frontend](#) / [store/store](#) / useAppDispatch

Function: useAppDispatch()

useAppDispatch(): *ThunkDispatch* <{ auctions: AuctionsState; auth:{ access_token: string | null; create_account_date: string | null; email: string | null; first_name: string | null; isAuthenticated: boolean; last_name: string | null; phone_number: string | null;}; categories: CategoriesState; userAuctions: MyAuctionsState;}, undefined, UnknownAction > & Dispatch < UnknownAction >

Defined in: [store/store.ts:22](#)

Returns

ThunkDispatch <{ auctions: AuctionsState; auth:{ access_token: string | null; create_account_date: string | null; email: string | null; first_name: string | null; isAuthenticated: boolean; last_name: string | null; phone_number: string | null;}; categories: CategoriesState; userAuctions: MyAuctionsState;}, undefined, UnknownAction > & Dispatch < UnknownAction >

docs\store\store\type-aliases\AppDispatch.md

[frontend](#)

[frontend](#) / [store/store](#) / AppDispatch

Type Alias: AppDispatch

AppDispatch = *typeof* store.dispatch

Defined in: [store/store.ts:19](#)

docs\store\store\type-aliases\AppStore.md

[frontend](#)

[frontend](#) / [store/store](#) / AppStore

Type Alias: AppStore

```
AppStore = typeof store
```

Defined in: [store/store.ts:20](#)

docs\store\store\type-aliases\RootState.md

[frontend](#)

[frontend](#) / [store/store](#) / RootState

Type Alias: RootState

```
RootState = ReturnType <*typeof* store.getState >
```

Defined in: [store/store.ts:18](#)

docs\store\store\variables\store.md

[frontend](#)

[frontend](#) / [store/store](#) / store

Variable: store

```
const store: EnhancedStore <{ auctions : AuctionsState; auth:{ access_token : string | null; create_account_date : string | null; email : string | null; first_name : string | null; isAuthenticated : boolean; last_name : string | null; phone_number : string | null;}; categories : CategoriesState; userAuctions : MyAuctionsState;}, UnknownAction , Tuple <[ StoreEnhancer <{}>, StoreEnhancer ]>>
```

Defined in: [store/store.ts:9](#)

docs\store\store\variables\useAppSelector.md

[frontend](#)

[frontend](#) / [store/store](#) / useAppSelector

Variable: useAppSelector

```
const useAppSelector: TypedUseSelectorHook < RootState > = useSelector
```

Defined in: [store/store.ts:23](#)

docs\store\thunks\AuctionsThunk\README.md

[frontend](#)

[frontend](#) / store/thunks/AuctionsThunk

store/thunks/AuctionsThunk

Functions

- [createNewAuctionThunk](#)
- [deleteAuctionThunk](#)
- [getAllAuctionsThunk](#)
- [getArchiveAuctionsThunk](#)
- [getAuctionCategoriesThunk](#)
- [getAuctionDetailsThunk](#)
- [getAuctionPhotoThunk](#)

- [getUserAuctionsThunk](#)
- [getUserOwnAuctionsThunk](#)
- [placeBidThunk](#)

docs\store\thunks\AuctionsThunk\functions\createNewAuctionThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / createNewAuctionThunk

Function: createNewAuctionThunk()

```
createNewAuctionThunk( auctionData ): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:178](#)

Thunk that creates a new auction on the backend with the given data. It takes an object with the following structure as an argument: { title: string, description: string, startPrice: number, startDate: string, endDate: string, categories: string[], images: { is_main: boolean; url: any }[] } It returns the result of the fetch call as a JSON object. If the response is not ok, it returns false. It also emits a "join" event to the socket with the id of the created auction.

Parameters

auctionData

any

The data of the new auction to be created.

Returns

The result of the fetch call as a JSON object, or false if the response is not ok.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\deleteAuctionThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / deleteAuctionThunk

Function: deleteAuctionThunk()

```
deleteAuctionThunk( auctionId ): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:87](#)

Thunk that deletes an auction by its id from the backend.

Parameters

auctionId

any

id of the auction to be deleted

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\getAllAuctionsThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / getAllAuctionsThunk

Function: getAllAuctionsThunk()

```
getAllAuctionsThunk(): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:11](#)

Thunk that fetches all auctions from the backend and returns the result as a JSON object.

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\getArchiveAuctionsThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / getArchiveAuctionsThunk

Function: getArchiveAuctionsThunk()

```
getArchiveAuctionsThunk(): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:66](#)

Thunk that fetches all archived auctions from the backend and returns the result as a JSON object.

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\getAuctionCategoriesThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / `getAuctionCategoriesThunk`

Function: `getAuctionCategoriesThunk()`

```
getAuctionCategoriesThunk(): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:227](#)

Thunk that fetches all categories from the backend and returns the result as a JSON object.

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

`dispatch`

`any`

`getState`

`any`

Returns

`Promise < any >`

`docs\store\thunks\AuctionsThunk\functions\getAuctionDetailsThunk.md`

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / `getAuctionDetailsThunk`

Function: `getAuctionDetailsThunk()`

```
getAuctionDetailsThunk( auctionId ): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:152](#)

Thunk that fetches an auction by its id from the backend and returns the result as a JSON object.

Parameters

`auctionId`

`any`

id of the auction to be fetched

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

`dispatch`

`any`

`getState`

`any`

Returns

`Promise < any >`

`docs\store\thunks\AuctionsThunk\functions\getAuctionPhotoThunk.md`

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / `getAuctionPhotoThunk`

Function: getAuctionPhotoThunk()

```
getAuctionPhotoThunk( imageUrl ): ( dispatch , getState ) => Promise < string >
```

Defined in: [store/thunks/AuctionsThunk.ts:108](#)

Thunk that fetches an image from the backend by its url and returns the result as a URL string.

Parameters

imageUrl

any

url of the image to be fetched

Returns

The result of the fetch call as a URL string.

```
( dispatch , getState ): Promise < string >
```

Parameters

dispatch

any

getState

any

Returns

Promise < string >

docs\store\thunks\AuctionsThunk\functions\getUserAuctionsThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / getUserAuctionsThunk

Function: getUserAuctionsThunk()

```
getUserAuctionsThunk(): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:47](#)

Thunk that fetches all auctions the current user is participating in from the backend and returns the result as a JSON object.

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\getUserOwnAuctionsThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / getUserOwnAuctionsThunk

Function: getUserOwnAuctionsThunk()

```
getUserOwnAuctionsThunk(): ( dispatch , getState ) => Promise < any >
```

Defined in: [store/thunks/AuctionsThunk.ts:26](#)

Thunk that fetches all auctions owned by the current user from the backend and returns the result as a JSON object.

Returns

The result of the fetch call as a JSON object.

```
( dispatch , getState ): Promise < any >
```

Parameters

dispatch

any

getState

any

Returns

Promise < any >

docs\store\thunks\AuctionsThunk\functions\placeBidThunk.md

[frontend](#)

[frontend](#) / [store/thunks/AuctionsThunk](#) / placeBidThunk

Function: placeBidThunk()

```
placeBidThunk( auctionId , bidAmount ): ( dispatch , getState ) => Promise < boolean >
```

Defined in: [store/thunks/AuctionsThunk.ts:243](#)

Thunk that places a bid on an auction with the given amount. It takes the id of the auction and the amount of the bid as arguments. It returns true if the response is ok, and false otherwise.

Parameters

auctionId

any

The id of the auction to place a bid on.

bidAmount

any

The amount of the bid to be placed.

Returns

True if the response is ok, false otherwise.

```
( dispatch , getState ): Promise < boolean >
```

Parameters

dispatch

any

getState

any

Returns

Promise < boolean >

docs\store\thunks\auth\AutoLoginThunk\README.md

[frontend](#)

[frontend](#) / [store/thunks/auth/AutoLoginThunk](#)

store/thunks/auth/AutoLoginThunk

Functions

- [autoLoginThunk](#)
- [getToken](#)

docs\store\thunks\auth\AutoLoginThunk\functions\autoLoginThunk.md

[frontend](#)

[frontend](#) / [store/thunks/auth/AutoLoginThunk](#) / autoLoginThunk

Function: autoLoginThunk()

```
autoLoginThunk(): ( dispatch , getState ) => Promise < boolean >
```

Defined in: [store/thunks/auth/AutoLoginThunk.ts:22](#)

Automatically logs in a user by retrieving the access token from the cookies. If the access token is invalid, logs out the user.

Returns

A promise resolved with true if the login was successful, false otherwise.

```
( dispatch , getState ): Promise < boolean >
```

Parameters

dispatch

any

getState

any

Returns

Promise < boolean >

docs\store\thunks\auth\AutoLoginThunk\functions\getToken.md

[frontend](#)

[frontend](#) / [store/thunks/auth/AutoLoginThunk](#) / getToken

Function: getToken()

```
getToken(): any
```

Defined in: [store/thunks/auth/AutoLoginThunk.ts:12](#)

Returns the access token from the cookies.

Returns

any

The access token.

docs\store\thunks\auth\ChangePasswordThunk\README.md

[frontend](#)

[frontend](#) / [store/thunks/auth/ChangePasswordThunk](#)

store/thunks/auth/ChangePasswordThunk

Functions

- [changePasswordThunk](#)

docs\store\thunks\auth\ChangePasswordThunk\functions\changePasswordThunk.

[frontend](#)

[frontend](#) / [store/thunks/auth/ChangePasswordThunk](#) / changePasswordThunk

Function: changePasswordThunk()

```
changePasswordThunk( params ): ( dispatch , getState ) => Promise < boolean >
```

Defined in: [store/thunks/auth/ChangePasswordThunk.ts:14](#)

Change password of user in backend

Parameters

params

object with oldPassword and newPassword

newPassword

any

oldPassword

any

Returns

- promise with result of change password operation

```
( dispatch , getState ): Promise < boolean >
```

Parameters

dispatch

any

getState

any

Returns

Promise < boolean >

docs\store\thunks\auth>LoginThunk\README.md

[frontend](#)

[frontend](#) / [store/thunks/auth/LoginThunk](#)

store/thunks/auth/LoginThunk

Functions

- [fetchUserData](#)
- [loginThunk](#)

docs\store\thunks\auth>LoginThunk\functions\fetchUserData.md

[frontend](#)

[frontend](#) / [store/thunks/auth/LoginThunk](#) / fetchUserData

Function: fetchUserData()

```
fetchUserData( accessToken ): Promise < Response >
```

Defined in: [store/thunks/auth/LoginThunk.ts:12](#)

Fetches user data from the backend by providing an access token

Parameters

accessToken

string

access token to be used for authentication

Returns

Promise < Response >

The result of the fetch call as a JSON object

docs\store\thunks\auth\LoginThunk\functions\loginThunk.md

[frontend](#)

[frontend](#) / [store/thunks/auth/LoginThunk](#) / loginThunk

Function: loginThunk()

```
loginThunk( The ): ( dispatch , getState ) => Promise < void >
```

Defined in: [store/thunks/auth/LoginThunk.ts:29](#)

Logs a user in by sending a POST request to the backend with the provided email and password

Parameters

The

email and password of the user to be logged in

email

any

password

any

Returns

A promise resolved when the login is successful or rejected

```
( dispatch , getState ): Promise < void >
```

Parameters

dispatch

any

getState

any

Returns

Promise < void >

docs\store\thunks\auth\RegisterThunk\README.md

[frontend](#)

[frontend](#) / [store/thunks/auth/RegisterThunk](#)

store/thunks/auth/RegisterThunk

Functions

- [isEmailTaken](#)
- [registerThunk](#)

docs\store\thunks\auth\RegisterThunk\functions\isEmailTaken.md

[frontend](#)

[frontend](#) / [store/thunks/auth/RegisterThunk](#) / isEmailTaken

Function: isEmailTaken()

`isEmailTaken(email): Promise < any >`

Defined in: [store/thunks/auth/RegisterThunk.ts:43](#)

Checks if the given email is already taken

Parameters

email

`string`

Email to check

Returns

`Promise < any >`

- True if the email is taken, false otherwise

docs\store\thunks\auth\RegisterThunk\functions\registerThunk.md

[frontend](#)

[frontend](#) / [store/thunks/auth/RegisterThunk](#) / registerThunk

Function: registerThunk()

`registerThunk(}): (dispatch , getState) => Promise < void >`

Defined in: [store/thunks/auth/RegisterThunk.ts:13](#)

Registers a user and logs them in if registration is successful

Parameters

`}`

email

`any`

name

`any`

password

`any`

phone

`any`

surname

`any`

Returns

`(dispatch , getState): Promise < void >`

Parameters

dispatch

`any`

getState

`any`

Returns

`Promise < void >`

docs\utils\formatCountdownPL\README.md

[frontend](#)

utils/formatCountdownPL

Functions

- [default](#)

docs\utils\formatCountdownPL\functions\default.md

Function: default()

```
default( iso ): string
```

Defined in: [utils/formatCountdownPL.ts:21](#)

Format countdown in Polish language.

Parameters

iso

```
string
```

ISO string of target date.

Returns

```
string
```

Formatted countdown string.