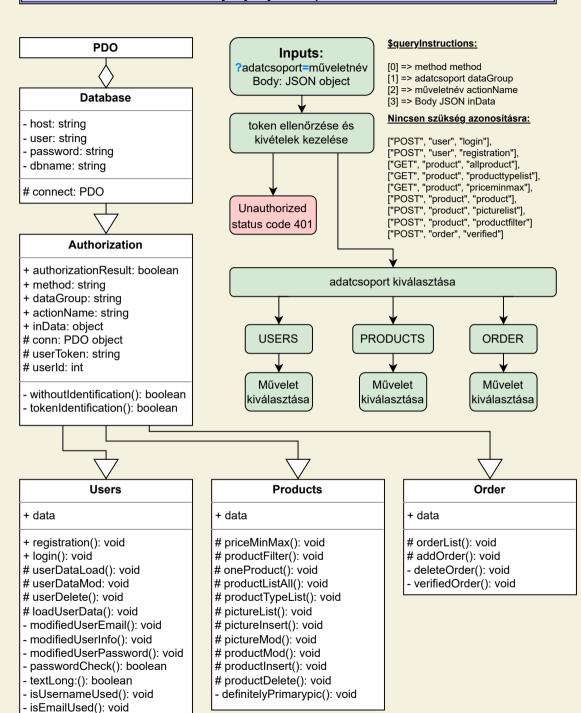
api.php felépítése



- loginInputsCheck(): boolean

userFavorite(): void

A jelszókat és generált tokeneket **SHA-512**-vel vannak kódolva. Az **SHA-512** egyedi 512 bites (32 bájtos) aláírást hoz létre egy szövegből.

A main.js metódusai

Page class

pageAddeventListeners drawMenuBg loadUserDatas (tokenId) pageLoad (inc) getBrowserVariables themeOptions pushDarkSwitchButton menuLogin (value) headFavoritButtonDraw headBasketButtonDraw

footerLinksOptions (value) generateProduct (tagName, attributes, value) lengthCheck (text, value, min, max, oneLength) countMarkdownPrice (price, markdown)

fistCharUpper (text) reloadPage

addProductToBasket (productGroup)

Basket class

UserDatasDraw basketButtonDraw basketAddEventListeners orderManagement orderButtonCheck createBasketList delBasketRow (row)

UserForm class

logOut
userLogin
userRegistration
regLoginSwitch(broght)
userDatasDrawInDisplay
regLogDraw
regResetValues
emailCheck(email)
userMod(key, value, min, max)
userPwdModOrDelete
(pwd, pwdNew, pwdNew2, mod)

Product class

LoadPriceMinMaxData
productsBasic
insertProductTypes
rangeMinMaxOptions
productSelected
productsList
drawOneProduct (product)
filterAddEventListeners
selectProductAddEventListeners
modifidProductPiece (value)
basketbuttonActiveOrNot
productSelectedFavoriteLoad (productid)

Favorites class

loadFavorites

Teszteléshez előre létrehozott felhasználók:

felhasználónév: admin00 jelszó: 123456 rang: admin admin@mikidani.probaljaki.hu

jelszó: gKXMvmJfPYpYZ9Kifkw3VxdUU40qNppH

felhasználónév: user00 jelszó: 123456 rang: user

user@mikidani.probaljaki.hu

jelszó: M6vlxEe1SZCU7qoVPKD9MDoY0XLqD6UD

mdshop@mikidani.probaljaki.hu

jelszó: GPG2kuEkiUn5tad13QdMcraamXM2ThNx

email belépések: https://webmail.nethely.hu/

Főbb oldalak:

http://web.mikidani.probaljaki.hu/mdshop/ http://web.mikidani.probaljaki.hu/mdshop/frontend/index.html http://web.mikidani.probaljaki.hu/mdshop/backend/index.php http://web.mikidani.probaljaki.hu/mdshop/api/api.php

MySql

users

id: Primary Key, int(11)

username: varchar(30)

email: varchar(50)

password: varchar(128)

info: text

rank: boolean

product

id: Primary Key, int(11)

typeid: int(11)

name: varchar(32)

text: text

price: int(11)

markdown: int(3)

rating: int(1)

type

id: Primary Key, int(11)

typename: varchar(30)

tokens

id: Primary Key, int(11)

userid: int(11)

token : varchar(128)

epochstart: int(10)

epochend: int(10)

product pic

id: Primary Key, int(11)

productid: int(11)

serverfilename: varchar(64)

text: varchar(256)

primarypic: boolean

favorites

id: Primary Key, int(11)

userid: int(11)

productid: int(11)

selectepoch: int(10)

product order

id: Primary Key, int(11)

userid: int(11)

productlist: JSON

totalprice: int(11)

orderepoch: int(10)

postalcode: int(11)

city: varchar(30)

designation: varchar(30)

designationtype: varchar(30)

designationnumber: int(11)

code: int(11)

verified (bollean)

REST API bemenetei:

API Query	Method	Token kell-e?	Admin jog ?	Várt bemenetek	Leírás
?user=login	POST			usernameoremail: string password: string	Helyes bejelentkezés esetén visszaküldi a token kódot.
?user=registration	POST			username:string email: string password: string	Ellenőrzi az adatok hosszúságát, illetve hogy regisztráltak-e már velük az adatbázisban. Ha nem akkor menti a felhasználót a users táblában.
?user=userdata	POST	igen		token: string usermail: string userinfo: string userpassword: string	Visszaadja a tokennel bejelentkezett felhasználó adatait.
?user=usermod	POST	igen	igen	token: tring useremail userinfo password	A felhasználó módosíthatja adatait. email, info, password
?user=userdelete	DELETE	igen	igen / nem	token: string password: string userid: number adminjog	Törli a felhasználót az adatbázisból.
?user=alluser	POST	igen	igen	token: string	Visszaadja az összes felhasználó adatait, kivéve jelszavak.
?user=favorites	POST	igen		token: string productid: int function: "get" "switsh"	GET: A felhasználó kedvenceit lehet lekérni. SWITCH: A listához lehet hozzáadni. Ha létezik akkor törli.

?product=allproduct	GET				Visszaadja az összes terméket.
?product=picturelist	POST			productid: int	Egy adott termékhez visszaadja a képek adatait.
?product=producttypelist	GET				Visszadja a product_type elemeit.
?product=pictureinsert	POST	igen	igen	token: string productid: int userfilename: string	Feltölti a képet a szerverre és regisztrálja az adatait a product_pic táblában.
?product=picturemod	POST	igen	igen	token: string pictureid: int productid: int function: "primarymod" "delete" "textmod" + pictext	Egy műveletnéven belül az összes kép adat módosítás itt hajtódik végre.
?product=productmod	POST	igen	igen	token: string productid: int function: "namemod" + productname "pricemod" + productprice "textmod" + producttext "typemod" + typeid "markdownmod" + markdown "instockmod" + instock	Egy műveletnéven belül az összes termék adat módosítás itt hajtódik végre.
?product=productfilter	POST			filtertype: string filtername: string filterminprice: number filtermaxprice: number filtermarkdown: boolean	A megadott feltételekkel szűri a termékeket és küldi vissza.
?product=priceminmax	GET				Visszaadja típusokra lebontva a legkisebb és legnagyobb árakat.
?product=productdelete	DELETE	igen	igen	token: string productid: number	Törli a teljes terméket képekkel együtt.
?order=list	POST	igen	igen	token: string	Visszaadja a megrendelések listáját időrendi sorrendben.
?order=add	POST	igen		token: string userid: number productlist: JSON totalprice: number postalcode: number city: string designation: string designationtype: string designationnumber: number	Rögzíti a megrendelést az adatbázisban.
?order=delete	DELETE	igen	igen	token: string orderid: number	Törli a megrendelést az adatbázisból.
?order=verified	POST			userid: number code: number	Visszaigazolja az emailban kiküldött megrendelést.