אוניברסיטת בן גוריון

הפקולטה להנדסה

המחלקה להנדסת מערכות מידע

סביבות פיתוח באינטרנט – עבודות בית 4+3, API

מגישים: עידו סולומון ת"ז 308111160

ליאור פרי ת"ז 203722814

Beers Inc.

<div dir="rtl">המוצר הנבחר - חנות בירות.</div>

ID	Method Name	HTTP Method	Parameters	Returns	Explanations
1	GetHottest5	GET	-	[{ "Beer Name": *beer_name*, "AlcoholPercentage": *perc* "Price": *price*, "Volume": *volume* },...]	Using GET because there is no information to deliver to the server or any wish to change any information.

2	Register	POST	{
			"Username":*"username"*,
			"Password": "*password*",
			"CountryID": *"countryID"*,
			"SecurityQuestions" : [
			{ "Question" : "*question1*",
			"Answer": "*answer1*" }, ...],
			"Categories": ["1","2"]
			}
				{
				"Succeeded": *boolean*,
				"Details": "*operation_details*"
				}
					Users make their own password retrieval question Returns a result code as int.
					Using POST because there are parameters being sent

3	GetCountryList	GET	-	String[]	Gets country data from XML Using GET because there is no information to deliver to the server or any wish to change any information.
4	IsUniqueUsernam	POST	{	{	True = unique

	e		"Username":*"username"* }	"Ans": *boolean*, "Details": *"operation_details"* }	False = already in use Using POST because there are parameters being sent

5	Login	POST	{
			 "Username":*"username".*
			 "Password":*"password"*
			 }
				Bool	Login cannot pass in GET
					 True = success
					 False=fail
					 Using POST because there are parameters being sent

6	ForgotPassword	POST	{
			 "Username": *"username"*
			 }
				[
				 "Question": *"question1"*
				 },
				 ...
				]
					Returns user-created questions
					 Using POST because there are parameters being sent

7	ValidateAnswer	POST	{
			 "Username": *"username"*,
			 "Question": *"question"*,
			 "Answer": *"Answer"*
			 }
				{
				 "CorrectAnswer": *boolean*,
				 "Password": *"password"*
				 }
					Returns empty string on failure, password string on success
					 Using POST because there are parameters being sent

8	GetNewProducts	GET	-	[{ "Beer Name": *beer_name*, "AlcoholPercentage": *perc* "Price": *price*, "Volume": *volume* },...]	Using GET because there is no information to deliver to the server or any wish to change any information.
9	GetRecommended	POST	{ "Username": *"username"*	[{	Recommended items are selected server-side

				"Beer Name": *beer_name*, "AlcoholPercentage": *perc* "Price": *price*, "Volume": *volume* },...]	Using POST because there are parameters being sent
			}		

10	Search	POST	{
			"BeerName" : "*BeerName*",
			"CategoryID" : "CategoryID",
			"AlcoholPercentage" : "*perc*",
			"Price" : "*Price*",
			"Volume" : "*Volume*",
			"" : "",
			
			}
				[
				{
				"Beer Name": *beer_name*,
				"AlcoholPercentage": *perc*
				"Price": *price*,
				"Volume": *volume*
				},...
				]
					Search is done server-side
					Using POST because there are parameters being sent

11	GetItemDetails	POST	{
			"BeerID": *beerId*
			
			}
				[
				{
				"Beer Name": *beer_name*,
				"AlcoholPercentage": *perc*
				"Price": *price*,
				"Volume": *volume*
				},...
				]
					ItemDetails expands Item
					Using POST because there are parameters being sent

12	GetPastOrders	POST	{
			"Username": "*username*"
			
			}
				[
				{
				"OrderID": ,
				"OrderDate":,
				"ShippingDate": ,
				"Currency":",
				"Total":, ,
				"Name": ,
					Order contains List of Items and other details such as order date and total
					Using POST because there are parameters being sent

				"AlcoholPercentage":, "Price":, "Volume":, "Quantity": }]	

13	MakeOrder	POST	{
			"Username":username,
			"Items": [
			{"BeerID": beerID,
			"Quantity": quantity},
			{"BeerID": beerID,
			"Quantity": quantity}, ...
			],
			"OrderDate": orderDate,
			"ShippingDate":
			shippingDate,
			"Currency": "USD",
			"CreditCard":
			creditCardNumber
			}
				OrderID	PaymentDetails includes CC number
					Return a new OrderID on success, -1 on failure
					Using POST because there are parameters being sent

14	IsInStock	POST	{
			"BeerID": *beerID*,
			"Quantity": *quantiy*
			
			}
				Bool	True = in stock, False = out of stock
					Using POST because there are parameters being sent

15	GetConversionRate	GET	-	Double	Gets USD to NIS conversion rate (Prices are stored on server in USD) Using GET because there

					is no information to deliver to the server or any wish to change any information.

16*	ManagerLogin	POST	{
			"Username":"username".
			"Password":"password"
			}
				Bool	Login cannot pass in GET
					True = success
					False=fail
					Using POST because there are parameters being sent

17*	GetOrderReport	POST	OrderID	OrderReport - Not used in this part of the assignment.	Using POST because there are parameters being sent

18*	UpdateItemDetails	POST	{
			"Username": "user",
			"CategoryID": CategoryID,
			"Name": "Name",
			"AlcoholPercentage": perce%,
			"Price": Price,
			"Volume": Volume
			}
				Bool	ItemDetails expands Item
					Using POST because there are parameters being sent

19*	AddItem	POST	{
			"Username": "user",
			"CategoryID": CategoryID,
			"Name": "Name",
			"AlcoholPercentage": perce%,
			"Price": Price,
			"Volume": Volume
			}
				Bool	Using POST because there are parameters being sent

20*	RemoveItem	POST	{ "Username""managerUser",	Bool	Using POST because there are parameters being sent

			"BeerID": *beerID*,}		

21*	AddUser	POST	{
			"Username"""*managerUser*",
			"AddedUsername": "*newUser*",
			"Password": "*passowrd*",
			"CountryID": *countryID*,
			"IsManager": *true/False*
			}
				Bool	User object contains all of a user's attributes
					Using POST because there are parameters being sent

22*	RemoveUser	POST	{
			"Username": *managerUser*,
			"UsernameDel": *username*,
			}
				Bool	Using POST because there are parameters being sent

23*	GetInventory	POST	{
			"Username":
			"*managerUser*",
			}
				[
				{
				"BeerID":*beerID*,
				"Name":" *beerName*",
				"Quantity": *quantity*
				},..
				]
					Int in tuple represents the quantity of the matching item in stock.
					Using GET because there is no information to deliver to the server or any wish to change any information.

24*	UpdateInventory	POST	{
			"Username":"*managerUser*",
			"BeerID":*beerID*,
			"Quantity":*quantity*
			}
				Bool	Inventory item contains all inventory listings and relevant data
					Using POST because there are parameters being sent

:הערות

- Cookie בצד לקוח שומר גם מועד התחברות אחרון וסל מוצרים.
- שמות השיטות מכילות את המילה items, כאשר הכוונה היא למוצרים שבחרנו (בירות).

- לקוח: ליאור פרי
  קשיים:
  - המרת צרכי המערכת לרשימת דרישות קלות להבנה.
  - התמקדות בפרטים הקטנים של המערכת, בניגוד ל"תמונה הגדולה".
- מתכנת: עדו סולומון
  קשיים:
  - פירוק דרישות הלקוח לפונקציות פשוטות.
  - קביעת חלוקת הפונקציונאליות בין צד שרת לצד לקוח.