שחר בראל 305420366, חן נחמני 301694865

API

ID	Method Name	HTTP Method	Parameters (Json example)/ URL parameter	Returns (Json example)	Explanation

1	/signIn	Post	{
			    "Username": "barelshahar",
			    "Password": "123456",
			    "FirstName": "shahar",
			    "LastName": "barel",
			    "Address": "klausner 4/6",
			    "City": "Beer Sheva",
			    "Country": "israel",
			    "Phone":"0524587717",
			    "Cellular":"0524587717",
			    "Mail":"barelshahar2@gmail.com",
			    "CreditCardNumber": "11111111",
			    "isAdmin": "false",
			    "Question": "ElementrySchool",
			    "Answer": "Sinai"
			}
				{
				  "status": "ok",
				  "response": []
				}
				
				Or
				
				{
				  "status": "failed",
				  "response": "failed to signIn- user name already exist"
				}
					SignIn cannot pass in GET

2	/addWatch	Post	{
			    "watchId": "qazaz",
			    "companyId": "4",
			    "description": "great watch",
			    "picPath": "C:/Users/shahar",
			    "categories": ["Luxury","sport"],
			    "stockAmount": "100",
			    "price": "50"
			}
				{
				  "status": "ok",
				  "response": []
				}
				Or
				{
				  "status": "failed",
				  "response": server response
				}
					A method used to add Watch to the database by an admin.
					Using POST because the server needs to insert data to his database.

3	/addCategory	Post	{
			  "category": "Sport"
			}
				{
				  "status": "ok",
				  "response": []
				}
				Or
				{
				  "status": "failed",
				  "response": server
				   response
				}
					A method used to add category to the database by an admin.
					Using POST because the server needs to insert data to his database.

4	/addCompany	Post	{
			  "company": "Rolex"
			}
				{
				  "status": "ok",
				  "response": []
				}
				Or
				{
				  "status": "failed",
				  "response": server
				   response
				}
					A method used to add company to the database by an admin.
					Using POST because the server needs to insert data to his database.

| 5 | /Login | Post | {
   "Username": "barelsha",
   "Password": "123456",
} | {
 "status": "ok",
 "response": [
  {
   "UserName":
   "barelsha"
  }
 ]
}
Or
{
 "status": "failed",
 "response": "failed to
  signIn- your password
  or Username not
  valid"
} | Login cannot pass in GET |

6	/listOfWatches	Get	------	{ "status": "ok", "response": [{ "WatchID": "BAVZ", "CompanyID": 4, "Description": "great watch", "CategoriesID": "sport,Luxury", "PicturePath": "D:/shahar/A", "Price": 50, "StokAmount": 108, "Date": "2017-04-05T00:00:00.000Z" }] }	A method used to get the list of all watches. Using GET because the client needs data from a specified resource (Watches). There is no need to insert data.

| 7 | /getWatchById | Get | Url parameter (/'id') | {
 "status": "ok",
 "response": [
  {
   "WatchID": "CAVD",
   "CompanyID": 4,
   "Description": "great watch",
   "CategoriesID": "Fox",
   "PicturePath": "D:/shahar/A",
   "Price": 50,
   "StokAmount": 100,
   "Date": "2017-06-08T00:00:00.000Z"
  }
 ]
}
Or
{
 "status": "failed",
 "response": "thre are no watches matching to this id"
} | Watches connected to server side, using GET because we have no data that flow to the server or update request. |

8	/isCategoryExist	Get	Url parameter (/'category')	{
				  "status": "ok",
				  "response": [
				    {
				      "CategoryID": 4,
				      "CategoryName": "Mens"
				    }
				  ]
				}
				Or
				{
				  "status": "failed",
				  "response": "there are no categories matching to this category"
				}
					A method used to check if category exist if adding Watch or User.
					Using GET because the client needs data from a specified resource (Categories). There is no need to insert data.

9	/isCompanyExist	Get	Url parameter (/'company')	{
				  "status": "ok",
				  "response": [
				    {
				      "CompanyId": 4,
				      "CompanyName": "Rolex"
				    }
				  ]
				}
				Or
				{
				  "status": "failed",
				  "response": "there are no companies matching to this category"
				}
					A method used to check if Company exist if adding Watch.
					Using GET because the client needs data from a specified resource (Companies). There is no need to insert data.

| 10 | /lastMonthWatches | Get | ------ | {
  "status": "ok",
  "response": [
    {
      "WatchID": "CAVD",
      "CompanyID": 4,
      "Description": "great watch",
      "CategoriesID": "Fox",
      "PicturePath": "D:/shahar/A",
      "Price": 50,
      "StokAmount": 100,
      "Date": "2017-06-08T00:00:00.000Z"
    }
  ]
} | A method used to retrieve list of new watches from last month. Using GET because we want to extract the news watches after login and there is no information to deliver to the server or any wish to change any information. |

11	/VerifyAnswer	Post	{
			   "Answer":"Sinai",
			 "Email":"barelshahar2@gmail.com",
			   "Question":"ElementrySchool"
			}
				{
				 "status": "ok",
				 "response": [
				  {
				   "Password": "123456"
				  }
				]
				}
					A method used to verify the answers to privacy questions. Using POST because we don't want the question and answer to be shown in the url.

12	/removeUser	Delete	{
			   "Username": "shahar"
			}
				{
				 "status": "ok",
				 "response": []
				}
				Or
				{
				 "status": "failed",
				 "response": server response
				}
					A method used to remove user from the users database by an admin. Using DELETE because the server needs to remove data from his database.

13	/removeWatch	Delete	{
			   "WatchID": "ZBVB"
			}
				{
				 "status": "ok",
				 "response": []
				}
				Or
				{
				 "status": "failed",
				 "response": "failed to
				   removeWatch - not exist"
				}
					A method used to remove Watch from the database by an admin. Using DELETE because the server needs to remove data from his database.

14	/isAdmin	Get	Url parameter (/'userName')	{
				  "status": "ok",
				  "response": "true" // or false
				}
					A method used to check if a user is an admin. Using GET because the client needs data from a specified resource (users). There is no need to insert data.

15	/categoriesList	Get	------	{
				  "status": "ok",
				  "response": [
				    {
				      "CategoryID": 1,
				      "CategoryName": "Sport"
				    },
				    {
				      "CategoryID": 2,
				      "CategoryName": "Luxury"
				    }
				  ]
				}
					A method used to retrieve list of categories. Using GET because we want to extract the categories and there is no information to deliver to the server or any wish to change any information.

16	/updateInventory	Put	{
			    "WatchID": "ZAVX",
			    "StokAmount": "-2"
			}
				{
				  "status": "ok",
				  "response": []
				}
				Or
				{
				  "status": "failed",
				  "response": "failed to updateInventory- Watch doesnt exist"
				}
					A method used to update stokAmount of a Watch. Using PUT because the server needs to update data of a record in his database.

| 17 | /listOfOrders | Get | Url parameter (/'userName') | {
  "status": "ok",
  "response": [
   {
    "OrderID": 4,
    "UserName": "chen",
    "OrderDate": "2017-06-08T00:00:00.000Z",
    "ShipmentDate": "2017-06-08T00:00:00.000Z",
    "Currency": "ILS",
    "TotalAmount": "3"
   }
  ]
}
Or
{
  "status": "failed",
  "response": "The username doesnt exist"
} | A method used to get the list of orders committed by the user.
Using GET because the client needs data from a specified resource (orders). There is no need to insert data. |