Mathaq user application API

# API base url :

http://zad.al-yasser.info/public/api/

Default method : POST

Notes: -

1. At pagination you can send (page) parameter in both methods (GET, POST)
2. The default application language is “en” unless you send it with “ar”, but there are some method that you should send it with “ar” or “en” only (will be notified)

# API authentication: -

There are two mandatory parameters you will send each time you use this api

1. **api\_email:** api\_user@authentication.com
2. **api\_password:** M\*y@H0tSu9\_o

note I will not ask you in this documentation to send those parameters as I mentioned here that it is mandatory to send them

**if authentication failed**

return {'status' : false, 'errNum': 500, 'msg' => 'Authentication failed'}

# User registration: -

**url** => http://zad.al-yasser.info/public/api/userSignUp

**params** => (full\_name, email, phone, country\_code, password, profile\_pic, image\_ext, longitude, latitude,invitation\_code, city\_id,country\_id,lang, reg\_id)

1- full\_name: (required | MIN: 3) -> represent user full name (string)

2- email: (required | email format) -> represent user email (string)

3- phone: (required | numeric) -> represent user phone (long || string)

4- password: (required | MIN: 8) -> represent user account password (string)

5- longitude: (optional) -> represent user location on map (double || string)

6- latitude: (optional) -> represent user location on map (double || string)

7- profile\_pic(optional) -> user profile pic string of data encoded in base64 (string)

8- image\_ext(optional) -> represent user profile pic extension (string)

9- invitation\_code (optional|exists) -> represent user inviter invitation code (string)

10- city\_id (required|numeric) -> represent city id(integer)

11- country\_id (required|numeric) -> represent country id(integer)

12- lang (Default:en) -> user application displayed language (string = "ar" || "en")

13 - country\_code (required) - > represent user phone country code

14 - reg\_id(required) - > represents user device firebase registration id

**If fails**

return {

'status' :false,

'errNum':errNum,

' msg' : msg

}

errNum =>

1 => 'Full Name is required',

2 => 'Full Name must be more than 3 characters',

3 => 'E-mail is required',

4 => 'Phone is required',

5 => 'Phone must be only digits',

6 => 'E-mail must be in e-mail format',

7 => 'Please locate your location on map',

8 => 'Password is required',

9 => 'Password must be more than 7 characters',

13 => 'Phone number is used before',

14 => 'E-mail is used before'

15 => 'You must determine profile picture extenstion',

16 => 'Wrong invitation\_code',

17 => 'city\_id is required',

18 => 'city\_id must be a number',

20 => 'Failed to upload image, please try again later'

21 => 'Proccess failed',

22 => 'country code is required',

23 =>'device register id is required',

24 => 'country\_id is required',

25 => 'country\_id must be a number',

Note:- if attribute (lang) is equal 'ar' these msgs will be returned in Arabic

**If success**

return {

'status' :true,

'errNum':0,

'user' : {

'user\_id' :user id,

'full\_name' :user fullName,

'invitation\_code' :user invitation\_code,

'user\_id' :user id,

'phone' :user phone,

'country\_code' :user phone country\_code,

'email' :user email,

'profile\_pic' :user image(url),

'longitude' :user longitude,

'latitude' :user latitude,

'city\_id' : city id,

'country\_id' : country id,

'city\_name' : city name

},

'msg' :'Added successfully'

}

# User login:-

**url** => http://zad.al-yasser.info/public/api/userLogin

**params** => (phone, password, reg\_id)

1- phone: (required) -> represents user phone

2- password: (required) -> represents user account password

3- reg\_id: (required) -> represents device new firebase id

**If fails**

**IF errNum =5**

return {

'status' :false,

'errNum':errNum,

'msg' :msg,

'user' : {

'full\_name' :user fullName,

'invitation\_code' :user invitation\_code,

'user\_id' :user id,

'phone' :user phone,

'status' : 0 => not activated, 1 activated,

'country\_code' :user phone country code,

'email' :user email,

'profile\_pic' :user image(url),

'longitude' :user longitude,

'latitude' :user latitude,

'city\_id' : city id,

'country\_id' : country id,

'city\_name' : city name

},

}

**else**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Phone is required',

2 => ' Wrong phone number (please add country code to the number)',

3 => 'Password is required',

4 => 'Wrong data',

5 => 'You need to activate your account'

6 => 'device number (reg\_id) is required'

**If success**

return {

'status' :true,

'errNum':0,

'user' : {

'user\_id' :user id,

'full\_name' :user full name,

'profile\_pic':user profile picture(url),

'status' :0 => not activated yet, 1 => activated,

'invitation\_code' : user invitation code,

'phone' : user phone,

'country\_code' : user phone country\_code,

'email' : user email,

'longitude' : user longitude,

'latitude' : user latitude,

'city\_id' : city id,

'city\_name' : city name

},

'msg':'Added successfully'

}

Note:- to login user must be activated

# User account activation:-

**url** => http://zad.al-yasser.info/public/api/activateUser

**params** => (user\_id)

1- user\_id: (required) -> represent registered user id

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Please provide us with user id',

2 => 'Failed to activate, please try again later'

**If success**

return {

'status' :true,

'errNum':0,

'msg' :'Activated successfully',

'user' : {

'user\_id' :user id,

'full\_name' :user full name,

'profile\_pic':user profile picture(url),

'status' :0 => not activated yet, 1 => activated,

'invitation\_code' : user invitation code,

'phone' : user phone,

'country\_code' : user phone country\_code

'email' : user email,

'longitude' : user longitude,

'latitude' : user latitude,

'city\_id' : city id,

'country\_id' : country id,

'city\_name' : city name

},

}

# Sending email API: -

**url** => [https://zad.al-yasser.info/public/api/sendMail](https://dev.al-yasser.info/zadapi/public/api/sendMail)

**params** => (email, lang)

1- email: (required|email) -> represent user email

2- lang : (required) -> represent application language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'User email is required',

2 => 'User email must be in email format',

3 => 'This email is not exist'**,**

4 =>'Failed to send, please try again later'

**If success**

return {

'status' :true,

'errNum':0,

'msg' :' Please check your mail to reset your password'

}

# User forget password:-

Make forget password? link redirect to this url =>

http://zad.al-yasser.info/public/api/forgetPass/{user\_id}

# Fetch providers' API: -

**url** => https://zad.al-yasser.info/public/api/getProviders

**params** => (cat\_id, user\_id, lang)

1. cat\_id:(optional)->represent accessed category id if it is null then all providers will be fetched
2. user\_id: (optional)->represent logged user id
3. lang: (default:en) -> represent application displayed language

**If fails**

return {

'status' :false,

'errNum':1,

'msg' :'There is no providers added yet!!'

}

**If success**

return {

'status' :true,

'providers':{

'total': 30,

'per\_page': 10,

'current\_page': 1,

'last\_page': 3,

'next\_page\_url': "https://zad.al-yasser.info/public/api/getProviders?page=2",

'prev\_page\_url': null,

'from': 1,

'to': 10,

'data': [

{

'provider\_id' : 1,

'full\_name' : "Provider1",

'profile\_pic' : "url",

'followers\_count': 0,

'provider\_rate' : 0,

'followFlag' : 0

},

{

'provider\_id' : 2,

'full\_name' : "Provider2",

'profile\_pic' : "url",

'followers\_count': 0,

'provider\_rate' : 0,

'followFlag' : 1

},

………..

],

},

'errNum':0,

'msg' : 'Retrieved successfully'

}

# Fetch categories API: -

**url** => https://zad.al-yasser.info/public/api/getCats

**params** => There no any needed params

**If fails**

return {

'status' :false,

'errNum':1,

'msg' :' There is no categories yet!!'

}

**If success**

return {

'status' :true,

'cats': [

{

'cat\_id' : 1,

cat\_name' : "قسم 1",

'cat\_img' : "url"

},

{

'cat\_id' : 2,

'cat\_name' : "Category2",

'cat\_img' : "url"

},

],

'errNum':0,

'msg' : 'Retrieved successfully'

}

# Provider screen API: -

**url** => http://zad.al-yasser.info/public/api/getProviderPage

**params** => (user\_id, cat\_id, provider\_id, lang)

1. user\_id: (sometimes|nullable|numeric) -> represent logged user id, if user not registered then = zero
2. cat\_id : (numeric) -> represent selected category id
3. provider\_id(required|numeric) -> represent selected provider id
4. lang(default:en) -> represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Invalid parameters' -> (must be number and can’t be null) ,

2 => 'Invalid provider id' -> (could be any number that doesn’t exist)

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Retrieved successfully ',

'provider' : {

"provider\_id": 1,

"full\_name": "Provider1",

"profile\_pic": "url",

"followers\_count": 0,

"provider\_rate": 0,

"followFlag": 1 if the logged user is following this provider else 0,

'allowed\_from\_time' : "9:00:00" start time of provider receive orders period,

'allowed\_to\_time' : "20:00:00" end time of provider receive orders period,

'receive\_orders' : 0 means provider doesn’t allow receiving any orders | 1= allow,

'current\_orders' : 0 = provider doesn’t allow receiving current orders | 1= allow,

'future\_orders' : 0 = provider doesn’t allow receiving future orders | 1= allow,

'avail\_date' : "yyyy-mm-dd hh:ii:ss" = last day provider can receive future orders ,

'delivery\_price' : 8.00

},

'providerCategories' : [

{

"cat\_id": 1,

"cat\_en\_name": "Category1",

"cat\_ar\_name": "قسم1",

"cat\_img": "url",

"flag ": 1 if it is the selected one, 0 if it is not

},

{

"cat\_id": 2,

"cat\_en\_name": "Category2",

"cat\_ar\_name": "قسم2",

"cat\_img": "url",

"flag ": 1 if it is the selected one, 0 if it is not

}

],

'meals' :{

'total': 30,

'per\_page': 10,

'current\_page': 1,

'last\_page': 3,

'next\_page\_url': "https://zad.al-yasser.info/public/api/getProviderPage?page=2",

'prev\_page\_url': null,

'form': 1,

'to': 10,

'data': [

{

"meal\_id": 1,

"meal\_name": "name1",

"meal\_rtp": "20", // real time price

"meal\_msrp": "20", // original price

"main\_image": "url",

"likes\_count": "0",

"meal\_rate": "0",

"likeFlag": "0",

},

{

"meal\_id": 2,

"meal\_name": "name2",

"meal\_rtp": "25", // real time price

"meal\_msrp": "25", //original price

"main\_image": "url",

"likes\_count": "10",

"meal\_rate": "3",

"likeFlag": "1",

},

………………..

],

},

"reportFlag": 1 means report before, 0 means didn't

}

Notes:-

1. In success case both “meals” and “providerCategories” could be an empty arrays
2. In “meals” section the attribute “likeFlag” indicates to if the user already like this meal, if it equal “0” then he didn’t
3. In “meals” section the attribute “meal\_rtp” represent the meal real time price

# Meal screen API: -

**url** => http://zad.al-yasser.info/public/api/getMealDetails

**params** => (meal\_id, user\_id, lang)

1. meal\_id: (required|numeric) -> represent visited meal id
2. user\_id: (optional|numeric) -> represent logged user
3. lang : (in:ar,en) -> represent app language

**If fails**

return {

'status' :false,

'errNum':1,

'msg' :'Invalid meal id'

}

**If success**

return {

"status": true,

"errNum": 0,

"meal": {

"meal\_id": 1,

"meal\_name": "تجربة وجبة",

"meal\_desc": "مجرد تجربة لوجبه علشان نشوف إيه الكلام",

"allowed\_qty": 20,

"main\_image": (url),

"video": (url),

"meal\_rtp": 60, //real time price

"meal\_msrp": 60, //original price

"meal\_rate": 0,

"likes\_count": 0,

"full\_name": "Provider1",

"provider\_id": 1,

"profile\_pic": "url",

"provider\_address": "string"

},

"userRate": 0 if user didn’t rate or there is no user logged in,

"attaches": [

{

"attach\_id": 1,

"attach\_path": "http://www.test.com/public/test/filename.jpg"

},

{

"attach\_id": 1,

"attach\_path": "http://www.test.com/public/test/filename2.jpg"

}

],

"comments": [

{

"full\_name": "mohamed",

"profile\_pic": "url",

"user\_id": 1,

"comment": "تجربة كومنت",

"id": 1

},

{

"full\_name": "mohamed",

"profile\_pic": "url",

"user\_id": 1,

"comment": "تجربة كومنت",

"id": 1

}

],

"comments\_count": 1,

"reportFlag": 1 means report before, 0 means didn't

"msg": "Retrieved successfully"

}

# User Order API: -

**url** => http://zad.al-yasser.info/public/api/addOrder

**params** => (lang, meals, provider\_id, user\_id, in\_future, address, delivery\_method\_id,

payment\_method\_id, delivery\_time, balance\_flag)

1. lang: (required|numeric) -> represent language of the application
2. meals : (required|numeric)-> represent array of meals data ([['meal\_id':1,'price':60,'qty':2,'discount':0],['meal\_id':1,'price':60,'qty':2,'discount':0]])if there is no any offer(discount) the discount will be equal zero
3. provider\_id(required|numeric)->represent selected provider id
4. user\_id: (required|numeric) -> represent user id
5. in\_future: (required|numeric|in:0,1) -> represent type of the order (in future = 1, now = 0)
6. address: (required|numeric) -> represent user address id
7. delivery\_method\_id: (required|numeric) ->represents used delivery method id
8. payment\_method\_id: (required|numeric)->represents used payment type id (COD=1,VISA=2)
9. delivery\_time: (required|in\_formate:Y-m-d H:i:s) -> represent expected delivery time
10. balance\_flag: (required|in:0,1) -> determin if the user will use his balance or not (0 => not use, 1 => use)

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

0 1 => 'There is no any meal in order',

2 => 'There is an error in some meal',

3 => 'There is an error in some meal price',

4 => 'There is an error in some meal count number',

5 => 'All data is required',

6 => 'lang field must be one of (ar, en)',

7 => 'in\_future field must be one of (0, 1)',

8 => 'delivery\_time field must be in format (Y-m-d H:i:s)',

9 => 'There is something wrong, please try again later',

10 => 'meal\_id can not be repeated',

11 => 'There is no enough quantaty for these meals',

12 => 'The provider doesn't receive orders',

13 => 'Meals must be an array',

14 => 'invalid address',

15 => 'balance\_flag must be 0 or 1',

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Process done successfully ',

'order\_code' :'string of 8 chars',

'order\_id' :Integer value,

}

NOTE:- This version of that API is not the last one, waiting for android department to start working on it to follow up with them to get the last version ( there is a potential of errors exist)

# Fetch User Orders: -

**url** => http://zad.al-yasser.info/public/api/orders

**params** => (user\_id, lang)

1. user\_id: (required|numeric) -> represent logged user id
2. lagn: (default:en) -> represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no orders yet',

2 => 'user\_id is required',

3 => 'user\_id must be a number'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :' Retrieved successfully',

'currentOrders' :[

{

'order\_id':1,

'order\_code':string,

'total\_value':100,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider name,

'profile\_pic':provider profile image(url),

'provider\_rate':provider rate(number),

' order\_status':Pending ,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23"

"is\_user\_complain\_provider": false or true

},

{

'order\_id':2,

'order\_code':string,

'total\_value':150,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider\_name,

' order\_status':Processing,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23",

"created\_date": "2017-05-03",

"created\_time": "10:38:23"

"is\_user\_complain\_provider": false

}

],

'futureOrders' :[

{

'order\_id':1,

'order\_code':string,

'total\_value':100,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider name,

'profile\_pic':provider profile image(url),

'provider\_rate':provider rate(number),

'order\_status':Pending,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23",

"created\_date": "2017-05-03",

"created\_time": "10:38:23"

"is\_user\_complain\_provider": false

},

{

'order\_id':2,

'order\_code':string,

'total\_value':150,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider\_name,

'order\_status':Processing,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23",

"created\_date": "2017-05-03",

"created\_time": "10:38:23"

"is\_user\_complain\_provider": false

}

],

'finishedOrders' :[

{

'order\_id':1,

'order\_code':string,

'total\_value':100,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider name,

'profile\_pic':provider profile image(url),

'provider\_rate':provider rate(number),

'order\_status':Done,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23",

"created\_date": "2017-05-03",

"created\_time": "10:38:23"

"is\_user\_complain\_provider": false or true

},

{

'order\_id':2,

'order\_code':string,

'total\_value':150,

'provider\_id':provider id,

'delivery\_id':delivery id ( will be equal zero if there is no delivery id for the order ),

'provider':provider\_name,

'order\_status':Failed to deliver,

"delivered\_at": "2018-01-14 14:35:29",

"delivered\_date": "2018-01-14",

"delivered\_time": "14:35:29",

"created\_at": "2017-05-03 10:38:23",

"created\_date": "2017-05-03",

"created\_time": "10:38:23"

"is\_user\_complain\_provider": false or true

}

]

}

# Fetch user Favorites and follows API: -

**url** => http://zad.al-yasser.info/public/api/getLikesFollowers

**params** => (user\_id, type)

1. user\_id: (required|numeric) -> represent logged user id
2. type : (required|numeric|in:1,2) -> 1 to get favorites & 2 to get follows

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'User\_id is required',

2 => 'User\_id must be a number',

3 => 'type is required',

4 => 'type must be a number',

5 => 'type must be 1 or 2',

6 => 'There is no data yet!!'

**If success**

**Case type = 1**

return {

'status' :true,

'errNum' :0,

'msg' :' Retrieved successfully',

'dataInfo' : {

'total': 30,

'per\_page': 10,

'current\_page':1,

'last\_page':3,

'next\_page\_url':"https://zad.al-yasser.info/public/api/getLikesFollowers?page=2,

'prev\_page\_url': null,

'from':1,

'to': 10,

'data':[

{

"meal\_id": 1,

"likes\_count": 6,

"main\_image": "url",

"meal\_name": "مطعم العنود",

"meal\_rate": 0,

"full\_name": provider name

},

{

"meal\_id": 1,

"likes\_count": 6,

"main\_image": "url",

"meal\_name": "مطعم العنود",

"meal\_rate": 0,

"full\_name": provider name

},

],

}

}

**Case type = 2**

return {

'status' :true,

'errNum' :0,

'msg' :' Retrieved successfully',

'dataInfo' : {

'total': 30,

'per\_page': 10,

'current\_page':1,

'last\_page':3,

'next\_page\_url':"https://zad.al-yasser.info/public/api/getLikesFollowers?page=2,

'prev\_page\_url': null,

'from':1,

'to': 10,

'data' :[

{

"provider\_id": 1,

"followers\_count": 6,

"profile\_pic": "url",

"full\_name": "مطعم العنود",

"provider\_rate": 0

},

{

"provider\_id": 2,

"followers\_count": 10,

"profile\_pic": "url",

"full\_name": "مطعم 2",

"provider\_rate": 0

},

]

}

}

# Meal likes and unlikes API: -

**url** => http://zad.al-yasser.info/public/api/mealLikeAndDislike

**params** => (user\_id, meal\_id, type)

1. user\_id: (required|numeric) -> represent logged user id
2. meal\_id: (required|numeric) -> represent meal id
3. type : (required|numeric|in:1,2) -> 1 to like the meal & 2 to unlike it

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'All fileds are required',

2 => 'All fileds must be a number',

3 => 'Type must be 1 or 2',

4 => 'Invalid user\_id',

5 => 'Invalid meal\_id',

6 => 'Your already liked this meal before',

7 => 'Process failed, please try again later',

8 => 'You did not like this meal before'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Proccess done successfully'

}

# Provider follows and unfollows: -

**url** => http://zad.al-yasser.info/public/api/providerFollowAndUnfollow

**params** => (user\_id, provider\_id, type)

1. user\_id: (required|numeric) -> represent logged user id
2. provider\_id: (required|numeric) -> represent meal id
3. type : (required|numeric|in:1,2) -> 1 to follow the provider & 2 to unfollow him

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'All fileds are required',

2 => 'All fileds must be a number',

3 => 'Type must be 1 or 2',

4 => 'Invalid user\_id',

5 => 'Invalid provider\_id',

6 => 'Your already followed this provider',

7 => 'Process failed, please try again later',

8 => 'You did not follow this provider before'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Proccess done successfully',

"num\_of\_followers" : 2

}

# Add comment API: -

**url** => http://zad.al-yasser.info/public/api/addComment

**params** => (user\_id, meal\_id, evaluation, comment, lang)

1. user\_id: (required|numeric|exists) -> represent logged user id
2. meal\_id: (required|numeric|exists) -> represent meal id
3. evaluation: (nullable|in:1,2, 3, 4, 5) -> represent user meal rate
4. comment: (required|min:3) -> represent user comment

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Comment, user\_id and meal\_id are required',

2 => 'evaluation, user\_id and meal\_id must be numbers',

3 => 'Comment must be not less than 3 chars',

4 => 'Invalid user\_id',

5 => 'Invalid meal\_id',

6 => 'evaluation must be between 1 and 5',

7 => 'There is something wrong, please try again later'

**If success**

return {

'status' :true,

'errNum' :0,

'overallRate' : meal overall rate after user rate ( = zero if he is not rate),

'msg' : 'Comment added '

}

# Add user address API: -

**url** => http://zad.al-yasser.info/public/api/addAddress

**params** => (user\_id, address, short\_address, longitude, latitude, lang)

1. user\_id: (required|numeric) -> represent logged user id
2. address: (required) -> represent user address
3. short\_address: (required) -> represent user short address
4. longitude: (optional) -> represent user location on map
5. latitude: (optional) -> represent user location on map
6. lang: (default:en)->represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Address filed is required',

2 => 'Short address filed is required',

3 => 'user\_id is required',

4 => 'user\_id must be a number',

5 => 'Proccess failed please try again later'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : ' Address has been added successfully ',

'addressDetail': {

"address\_id": 59,

"address": "String",

"short\_address": "String",

"longitude": "could be null",

"latitude": "could be null"

}

}

# Fetch user addresses API:-

**url** => http://zad.al-yasser.info/public/api/getUserAddress

**params** => (user\_id, lang)

1. user\_id: (required|numeric) -> represent logged user id
2. lang: (default:en)->represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'user\_id is required',

2 => 'user\_id must be a number',

3 => 'There is no addresses yet'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully ',

"addresses": [

{

"address\_id": 1,

"user\_id": 1,

"short\_address": "عنوان تجريبى1",

"address": "مجرد عنوان تجريبى لتسهيل تجربة البرنامج",

"longitude": "could be null",

"latitude": "could be null "

},

{

"address\_id": 20,

"user\_id": 1,

"short\_address": "عنوان تجريبى20",

"address": "مجرد عنوان تجريبى لتسهيل تجربة البرنامج",

"longitude": "could be null ",

"latitude": "could be null "

}

]

}

# Delete user address API: -

**url** => http://zad.al-yasser.info/public/api/deleteUserAddress

**params** => (address\_id, lang)

1. address\_id: (required|numeric) -> represent logged user id
2. lang: (default:en)->represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'address\_id is required',

2 => 'address\_id must be a number',

3 => 'address\_id is not exist',

4 => 'Proccess failed, please try again later'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Deleted successfully'

}

# Make Provider Report API: -

**url** => http://zad.al-yasser.info/public/api/providerReport

**params** => (provider\_id, lang)

1. provider \_id: (required|numeric) -> represent reported provider id
2. user\_id: (required|numeric) -> represent logged user id
3. lang: (default:en)->represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'provider\_id and user\_id are required',

2 => 'provider\_id and user\_id must be a number',

3 => 'provider\_id or user\_id is not exists',

4 => 'Failed to report, please try again later',

5 => 'You already reported this provider'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Reported successfully'

}

# Provider evaluation API

**url** => http://zad.al-yasser.info/public/api/provider\_evaluate

**params** => (quality,autotype,packing,maturity, askAgain,comment, user\_id, order\_id, lang)

'quality' => 'required|in:1,2,3,4,5',

'autotype' => 'required|in:1,2,3,4,5',

'packing' => 'required|in:1,2,3,4,5',

'maturity' => 'required|in:1,2,3,4,5',

'askAgain' => 'required|in:1,2,3,4,5',

'comment' => 'max:140',

'user\_id' => 'required',

'order\_id' => 'required',

'lang' => default:en

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'All the evaluation fileds are required',

2 => 'Evaluate values must be between 1 and 5',

3 => 'Failed to evaluate, please try again later'

4 => 'Comment can not be more than 140 characters'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : ' Evaluated successfully'

}

# Delivery evaluation API

**url** => http://zad.al-yasser.info/public/api/delivery\_evaluate

**params** => (arrival, inTime, attitude,comment, user\_id, order\_id, lang)

'arrival' => 'required|in:1,2,3,4,5',

'inTime' => 'required|in:1,2,3,4,5',

'attitude' => 'required|in:1,2,3,4,5',

'comment' => 'max:140',

'user\_id' => 'required',

'order\_id' => 'required',

'lang' => default:en

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'All the evaluation fileds are required',

2 => 'Evaluate values must be between 1 and 5',

3 => 'Failed to evaluate, please try again later'

4 => 'Comment can not be more than 140 characters'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : ' Evaluated successfully'

}

# Add attachment API:-

**url** => http://zad.al-yasser.info/public/api/addAttach

**params** => (image\_str, image\_ext, attach\_no, lang)

1. image \_str: (required) -> represent image string of data
2. image\_ext: (required)->represent image extension
3. attach\_no: (required|numeric)->represent attachments group number
4. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Image is required',

2 => 'image\_ext is required',

3 => 'attach\_no must be number',

4 => 'Failed to upload please try again later',

5 => 'Failed to add the file',

6 => 'attach\_no is required'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'uploaded successfully' ,

'file\_id' :1, // represent unique integer for the uploaded file

'attach\_no' :5 //represent the attached group number

}

Note:- in the first uploaded file you will send parameter (attach\_no) with zero value, then you will receive a new attach\_no value at the response, that value you will send in all other attached files

# Delete attachment API:-

**url** => http://zad.al-yasser.info/public/api/deleteAttach

**params** => (file\_id, attach\_no, lang)

1. file\_id: (required|numeric) -> represent file number
2. attach\_no: (required|numeric)->represent attachments group number
3. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'file\_id is required',

2 => 'file\_id must be a number',

3 => 'attach\_no is required',

4 => 'attach\_no must be a number',

5 => 'Failed to remove the file, please try again later'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Removed successfully' ,

'attach\_no' :5 //represent the attached group number(will be 0 if there is no any files in that number)

}

# Add complain API:-

**url** => http://zad.al-yasser.info/public/api/complain

**params** => (subject\_id, order\_id, actor\_id,files, extentions, types,complain, complaint, app,lang)

1. subject\_id: (required|numeric) -> represent the subject (provider or delivery) id
2. order\_id: (required|numeric|exists)-> represent order number
3. actor\_id: (required|numeric|exists) -> represent actor (user, provider, delivery) number
4. files: ( optional| array) -> represent array of base64 encoded files strings [sting of image, string of image, string of video, ...]
5. extentions: ( optional|array) -> represent array of attached files extensions [jpg, png, mp4, ….]
6. types: (optional|array) -> represent type of attached file ['image', 'image', 'video',.....]
7. complain: (required) -> represent the user text complain
8. complaint: (required|in:provider,delivery) -> determine if the complain is for a provider or a delivery
9. app: (required|in:user,provider) -> represents the used app to add complain it must be equal 'user' in this application
10. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Invalid parameters',

2 => 'Process failed, please try again later'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Process done successfully'

}

# Fetch timeline data API: -

**url** => http://zad.al-yasser.info/public/api/timeLine

**params** => (user\_id, lang)

1. user\_id: (required|numeric|exists) -> represent user number
2. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Invalid parameters',

2 => 'There is no any data'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"notifications": {

"total": 52,

"per\_page": 10,

"current\_page": 1,

"last\_page": 6,

"next\_page\_url": "https://zad.al-yasser.info/devpublic/api/timeLine?page=2",

"prev\_page\_url": null,

"from": 1,

"to": 10,

"data": [

{

"meal\_id": 1,

"created\_at": "2017-05-05 02:33:45",

"notif\_desc": "مجرد تجربه رقم 1",

"meal\_main\_img": "http://zad.al-yasser.info/public/mealsImages/main.jpg",

"meal\_meal\_rtp": meal real time price,

"meal\_meal\_msrp": meal original price,

"provider\_image": "http://zad.al-yasser.info/public/userProfileImages/mac.png",

"provider\_name": "مطعم العنود",

"meal\_name": "وجبة1",

"meal\_rate": 2,

"cat\_img": "http://zad.al-yasser.info/public/categoriesImages/lunch\_ic.png",

"cat\_id": 1,

"cat\_name": "وجبات الغداء"

},

{

"meal\_id":2,

"created\_at": "2017-05-05 02:33:45",

"notif\_desc": "مجرد تجربه رقم 2",

"meal\_main\_img": "http://zad.al-yasser.info/public/mealsImages/main.jpg",

"meal\_meal\_rtp": meal real time price,

"meal\_meal\_msrp": meal original price,

"provider\_image": "http://zad.al-yasser.info/public/userProfileImages/mac.png",

"provider\_name": "مطعم العنود",

"meal\_name": "وجبة1",

"meal\_rate": 2,

"cat\_img": "http://zad.al-yasser.info/public/categoriesImages/lunch\_ic.png",

"cat\_id": 1,

"cat\_name": "وجبات الغداء"

},

…………..

]

}

}

# Edit profile API: -

**url** => http://zad.al-yasser.info/public/api/editProfile

**params** => (user\_id,full\_name, email, phone, country\_code, password, profile\_pic, image\_ext, longitude, latitude, city\_id, country\_id, lang)

1- user\_id: (required) -> represent logged user id(int)

2- full\_name: (required | MIN: 3) -> represent user full name (string)

3- email: (required | email format|unique) -> represent user email (string)

4- password: (optional | MIN: 8) -> represent user account password (string)

5- longitude: (optional) -> represent user location on map (double || string)

6- latitude: (optional) -> represent user location on map (double || string)

7- profile\_pic(optional) -> user profile pic string of data encoded in base64 (string)

8- image\_ext(optional) -> represent user profile pic extension (string)

9- city\_id: (required|numeric) -> represents city id

10- country\_id: (required|numeric) -> represents country id

11- lang (default:en) -> user application displayed language (string = "ar" || "en")

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Full Name is required',

2 => 'Full Name must be more than 3 characters',

3 => 'E-mail is required',

6 => 'E-mail must be in email format',

9 => 'Password must be more than 7 characters',

13 => 'Phone number is used before',

14 => 'E-mail is used before',

15 => 'You must determine profile picture extension',

16 => 'user\_id is required',

17 => 'user\_id is not exists',

18 => 'city is required',

19 => 'city\_id must be a number',

20 => 'Failed to update please try again later',

22 => 'country code is required',

23 => 'country is required',

24 => 'country\_id must be a number',

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Process done successfully' ,

'user' : {

'full\_name' :user fullName,

'user\_id' :user id,

'phone' :user phone,

'email' :user phone,

"longitude" : "123456.123456",

"latitude": "7890.7890",

'profile\_pic' :user image(url),

'city\_id' : city id,

'country\_id' : country id,

},

}

# Edit phone API: -

**url** => http://zad.al-yasser.info/public/api/edit\_phone

**params** => (actor\_id, country\_code, phone, app, lang)

1- actor\_id: (required) -> represents the phone owner id (could be provider, user or delivery man)

2- country\_code: (required) -> represents phone country code

3- phone: (required|unique)-> represents new phone number

4- app: (required |in: users, providers, deliveries) -> represents the used app

5- lang (default:en) -> user application displayed language (string = "ar" || "en")

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'All fields are required',

2 => 'app must be in (users,providers, deliveries)',

3 => 'Failed to update, please try again later',

4 => 'Phone number is already exist'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Phone updated successfully'

}

Note:- this api used for all applications

# Get countries API: -

**url** => http://zad.al-yasser.info/public/api/countries

**params** => (lang)

1. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no data'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"countries": [

{

"country\_id": 1,

"country\_name": "string name",

"country\_name": ex:- 0966

},

{

"country\_id": 1,

"country\_name": "string name",

"country\_name": ex:- 0966

},

…………..

]

}

# Get country cities API: -

**url** => http://zad.al-yasser.info/public/api/countryCities

**params** => (country\_id, lang)

1. country\_id: (required)->represent selected country id
2. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no data',

2 => 'country\_id is required'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"cities": [

{

"city\_id": 1,

"city\_name": "string name",

"country\_code": ex: 0966

},

{

"city\_id": 2,

"city\_name": "string name",

"country\_code": ex: 0966

},

…………..

]

}

# Get cities API: -

**url** => http://zad.al-yasser.info/public/api/cities

**params** => (lang)

1. lang: (default:en)->represent app displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no data'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"cities": {

"country name": [

{

"city\_id": 1,

"city\_name": "string name",

"country\_code":0966

},

{

"city\_id": 2,

"city\_name": "string name",

"country\_code":0966

},

………..

],

"country name": [

{

"city\_id": 1,

"city\_name": "string name",

"country\_code":0966

},

{

"city\_id": 2,

"city\_name": "string name",

"country\_code":0966

},

………..

],

………..

}

# Search API: -

**url** => http://zad.al-yasser.info/public/api/search

**params** => (name,city,cat\_id,delivery\_id,meals\_count,date,distance,user\_longitude,user\_latitude,type,lang)

1. name => (optional)->represent name of provider or meal
2. city => (required)->represent the selected city id
3. cat\_id => (optional)->represent provider or meal category
4. delivery\_id => (optional)->represent provider delivery method id
5. meals\_count => (optional)->represent provider meals count
6. date => (optional|required if time exist)->represent provider available date
7. distance => (optional)->represent the distance that user provided
8. user\_longitude => (required if distance exist)->represent logged user longitude
9. user\_latitude => (required if distance exist)->represent logged user latitude,
10. type =>(in:provider,meal)->represents desired search result providers or meals

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no result for your search',

2 => 'You must determine the date if you search with time',

3 => 'user longitude and latitude is required',

4 => 'distance must be in number',

6 => 'date must be in format yyyy-mm-dd',

7 => 'Please select city'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"result": [

{

"id": meal or provider id

"name": meal or provider name

"rate": meal or provider rate,

"img": "url",

"likes\_count": it "type" equal 'provider' will be -1,

"followers\_count": it "type" equal 'meal' will be -1,

"type": "meal or provider"

"price": "60.00" // this filed is zero in case of provider

},

{

"id": meal or provider id

"name": meal or provider name

"rate": meal or provider rate,

"img": "url",

"likes\_count": it "type" equal 'provider' will be -1,

"followers\_count": it "type" equal 'meal' will be -1,

"type": "meal or provider",

"price": "0" // this filed is zero in case of provider

},

………..

]

}

# Prepare Search API: -

**url** => http://zad.al-yasser.info/public/api/prepareSearch

**params** => (lang)

1. lang: (default:en) -> represent application displayed language

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no data'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"cities": [

{

"city\_id": city id

"city\_name": name of city

},

{

"city\_id": city id

"city\_name": name of city

},

………

],

"deliveries": [

{

"method\_id": delivery id

"delivery\_name": name of delivery

},

{

"method\_id": delivery id

"delivery\_name": name of delivery

},

………

],

"cats": [

{

"cat\_id": cat id

"cat\_name": name of cat

},

{

"cat\_id": cat id

"cat\_name": name of cat

},

………

]

}

# Prepare Payment API: -

**url** => http://zad.al-yasser.info/public/api/preparePayment

**params** => (lang, user\_id)

1. lang: (default:en) -> represent application displayed language
2. user\_id: (required) -> represent user id
3. provider\_id: (required) -> represent provider id

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no data',

2 => 'user\_id is required'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"deliveries": [

{

"method\_id": delivery id

"delivery\_name": name of delivery

},

{

"method\_id": delivery id

"delivery\_name": name of delivery

},

………

],

"addresses": [

{

"address\_id": address id,

"user\_id": 1,

"short\_address": تجربه,

"address": تجربه,

"longitude": 111.2342,

"latitude ": 234.3424

},

{

"address\_id": address id,

"user\_id": 1,

"short\_address": تجربه,

"address": تجربه,

"longitude": 111.2342,

"latitude ": 234.3424

},

………

],

"is\_provider\_allow\_future\_orders": 1

}

# SMS API: -

**url** => http://zad.al-yasser.info/public/api/sms

**params** => (lang, phone)

1. lang: (default:en) -> represent application displayed language
2. phone: (required) -> represent user phone

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'Failed to send the message',

2 => 'Phone is required'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Message sent successfully' ,

'code' : 'sent code'

}

# Prepare sign up API: -

**url** => http://zad.al-yasser.info/public/api/prepareSignUp

**params** => (lang)

1. lang: (default:en) -> represent application displayed language

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

'countries' : [

{

"country\_id": city id,

"country\_name": name of country,

"country\_code": country code

},

{

"country\_id": country id,

"country\_name": name of country,

"country\_code": country code

},

…….

]

}

# Get order details API: -

**url** => http://zad.al-yasser.info/public/api/orderDetails

**params** => (lang, order\_id)

1. lang: (default:en) -> represent application displayed language
2. order\_id: (required|numeric)-> represents order id

**If success**

return {

'status' :true,

'errNum' :0,

'msg' : 'Retrieved successfully' ,

"header": {

"order\_code": "42345454 ",

"total": "180", //note that total value is the order price + order delivery price

"delivery\_price": 20,

"app\_value": 20,

"marketer\_delivery\_value": 20,

"marketer\_provider\_value": 20,

"user\_address": "test address",

"user\_longitude": 33.43342,

"user\_latitude": 33.33345,

"user\_phone": "1234567890",

"user\_email": "test1@test.com",

"delivered\_at": "0000-00-00 00:00:00",

"payment\_method": "Cash On Delivery",

"delivery\_method": "Aramex",

"provider\_name": "مطعم العنود",

"user\_name": "user name" //To use in provider application

"order\_type": " مستقبلى",

"order\_status": " جارى التسليم",

"status\_id": 3,

"order\_date": "2017-05-25"

},

'details' : [

{

"meal\_name": meal name,

"meal\_price": 60,

"meal\_desc":Description of meal,

"discount": 0,

"qty": 2

},

{

"meal\_name": meal name,

"meal\_price": 60,

"meal\_desc":Description of meal,

"discount": 0,

"qty": 2

},

…….

],

"app\_percentage": 2 // to use in provider application

"order\_rate": 3 // note if the order status was not delivered then this attribute will be equal ""

}

# Get provider meals: -

**url** => http://zad.al-yasser.info/public/api/providerMeals

**params** => (provider\_id,cat\_id,user\_id,lang)

1. lang: (default:en) -> represent application displayed language
2. provider\_id: (required|numeric)-> represent provider id
3. cat\_id: (optional|numeric) -> represent category id
4. user\_id: (optional|numeric)-> represent user id

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Retrieved successfully ',

'meals' :{

'total': 30,

'per\_page': 10,

'current\_page': 1,

'last\_page': 3,

'next\_page\_url': "https://zad.al-yasser.info/public/api/providerMeals?page=2",

'prev\_page\_url': null,

'form': 1,

'to': 10,

'data': [

{

"meal\_id": 1,

"meal\_name": "name1",

"meal\_desc": "desc1",

"meal\_rtp": "20", // real time price

"meal\_msrp": "20", // original price

"main\_image": "url",

"likes\_count": "0",

"meal\_rate": "0",

"likeFlag": "0" 0 if user is not logged in or doesn't like the meal,

},

{

"meal\_id": 2,

"meal\_name": "name2",

"meal\_desc": "desc2",

"meal\_rtp": "25", // real time price

"meal\_msrp": "25", //original price

"main\_image": "url",

"likes\_count": "10",

"meal\_rate": "3",

"likeFlag": "1" 1 if user likes the meal,

},

………………..

],

}

}

# Prepare profile API: -

**url** => http://zad.al-yasser.info/public/api/getEditProfile

**params** => (user\_id, lang)

1- user\_id: (required) -> represent userid

2- lang (Default:en) -> user application displayed language (string = "ar" || "en")

**If fails**

return {

'status' :false,

'errNum':errNum,

' msg' : msg

}

errNum =>

1 => user\_id is required',

2 => user\_id must be a number',

3 => 'There is no data'

**Note:- if attribute (lang) is equal 'ar' these msgs will be returned in Arabic**

**If success**

return {

'status' :true,

'errNum':0,

'msg' :'',

'userData' : {

'user\_id' :user id,

'full\_name' :user full name,

'profile\_pic':user profile picture(url),

'status' :0 => not activated yet, 1 => activated,

'invitation\_code' : user invitation code,

'phone' : user phone,

'country\_code' : user phone country\_code

'email' : user email,

'longitude' : user longitude,

'latitude' : user latitude,

'city\_id' : city id,

'country\_id' : country id,

'city\_name' : city name

},

"countries": [

{

"country\_id": 1,

"country\_name": "Saudi Arabia",

"chosen": 0,

"country\_code": 966

},

{

"country\_id": 2,

"country\_name": "Egypt",

"chosen": 0,

"country\_code": 966

}

],

"cities": [

{

"city\_id": 1,

"city\_name": "Gada",

"chosen": 1

},

{

"city\_id": 2,

"city\_en\_name": "El Ryad",

"chosen": 0

},

{

"city\_id": 3,

"city\_en\_name": "Cairo",

"chosen": 0

},

{

"city\_id": 4,

"city\_name": "Alexandria",

"chosen": 0

}

],

"cats": [

{

"cat\_id": 1,

"cat\_name": "Launch",

"chosen": 1

},

{

"cat\_id": 2,

"cat\_name": "Dinner",

"chosen": 1

},

{

"cat\_id": 3,

"cat\_en\_name": "Breakfast",

"chosen": 1

},

……………..

]

}

# Visitor Order API: -

**url** => http://zad.al-yasser.info/public/api/addVisitorOrder

**params** => (lang, meals, provider\_id, in\_future, delivery\_method\_id,

payment\_method\_id, delivery\_time, visitor\_phone, visitor\_country\_code, visitor\_address, visitor\_address\_lat, visitor\_address\_long, visitor\_email,visitor\_id)

1. lang: (required|numeric) -> represent language of the application
2. meals : (required|numeric)-> represent array of meals data ([['meal\_id':1,'price':60,'qty':2,'discount':0],['meal\_id':1,'price':60,'qty':2,'discount':0]])if there is no any offer(discount) the discount will be equal zero
3. provider\_id(required|numeric)->represent selected provider id
4. in\_future: (required|numeric|in:0,1) -> represent type of the order (in future = 1, now = 0)
5. delivery\_method\_id: (required|numeric) ->represents used delivery method id
6. payment\_method\_id: (required|numeric)->represents used payment type id (COD=1,VISA=2)
7. delivery\_time: (required|in\_formate:Y-m-d H:i:s) -> represent expected delivery time
8. visitor\_phone: (required|unique) -> represents visitor phone number
9. visitor\_country\_code: (required) -> represents visitor country code
10. visitor\_address:(required|string) -> represents visitor address string
11. visitor\_address\_lat:(required) -> represents visitor address location on map
12. visitor\_addres\_long:(required)-> represents visitor address location on map
13. visitor\_email: (required|email|unique) -> represents visitor email
14. visitor\_id: (optional|could be zero)-> represents visitor id in case of he added an order before

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'There is no any meal in order',

2 => 'There is an error in some meal',

3 => 'There is an error in some meal price',

4 => 'There is an error in some meal count number',

5 => 'All data is required',

6 => 'lang field must be one of (ar, en)',

7 => 'in\_future field must be one of (0, 1)',

8 => 'delivery\_time field must be in format (Y-m-d H:i:s)',

9 => 'There is something wrong, please try again later',

10 => 'meal\_id can not be repeated',

11 => 'There is no enough quantaty for these meals',

12 => 'The provider doesn\'t receive orders',

13 => 'Meals must be an array',

14 => 'invalid address',

15 => 'Invalid e-mail format',

16 => 'This e-mail is used before',

17 => 'This phone number is used before'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Process done successfully ',

'order\_code' :'string of 8 chars',

'order\_id' :Integer value,

'visitor\_id' :Integer value,

'visitor\_pic' :default user profile image link,

'visitor\_name' :name set by application,

}

# Cancel order API: -

**url** => http://zad.al-yasser.info/public/api/userCancelOrder

**params** => (lang, user\_id, order\_id)

1. lang: (required|numeric) -> represent language of the application
2. user\_id : (required|numeric)-> represent user id
3. order\_id: (required|numeric)->represent selected order id

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'order\_id is required',

2 => 'user\_id is required',

3 => 'Failed to cancel the order, try again',

4 => 'Sorry it is not your order to cancel',

5 => 'Sorry you can't cancel order with delivery man'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Order has been canceled '

}

# User balance API: -

**url** => http://zad.al-yasser.info/public/api/userBalance

**params** => (lang, user\_id)

1. lang: (required|numeric) -> represent language of the application
2. user\_id : (required|numeric)-> represent user id

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'user\_id is required'

**If success**

return {

"status": true,

"errNum": 0,

"msg": "",

"total\_balance": 12561,

"balance\_details": [

{

"total\_value": "14149",

"day": "2017-10-16",

"status": 6,

"full\_text": "14149 SR from refused order at 2017-10-16"

}

],

"usedCredit": 0,

"invitationCredits": 0,

"withdrawed\_balance": 0,

"bank\_name":””,

"bank\_phone":””,

"account\_num":””,

"bank\_username": “”

}

# User balance withdraw API: -

**url** => http://zad.al-yasser.info/public/api/userWithdraw

**params** => (lang,user\_id,value,bank\_name,name,account\_num,phone)

1. lang: (required|numeric) -> represent language of the application
2. user\_id : (required)-> represent user id
3. value: (required|numeric)->represents requested balance value
4. bank\_name: (required) -> represents bank name
5. account\_num: (required)-> represents user account number
6. phone : (required)-> represents user phone

**If fails**

return {

'status' :false,

'errNum':errNum,

'msg' :msg

}

errNum =>

1 => 'user\_id is required',

2 => 'value is required',

3 => 'value must be a nubmer',

4 => 'Bank name is required',

5 => 'Name is required',

6 => 'Account number is required',

7 => 'Phone number is required',

8 => 'Process failed, please try again',

9 => 'You have a pending request, you can\'t add that request'

**If success**

return {

'status' :true,

'errNum' :0,

'msg' :'Added successfully'

}