**Sallahnow Api (teachnicians) :-**

Technician authenticition part :

Register new technician api :

url : host/technicians/register

method : POST

Response type : JSON

Body :

* tech\_name
* tech\_email
* tech\_mobile
* tech\_tel
* tech\_password
* tech\_identification
* tech\_birth
* country\_id
* state\_id
* area\_id
* tech\_address

Response :

On success :-

Status code = 100

Data = [ access\_token : token, token\_type : bearer, expires\_in : int(0), status\_code : int(100)]

On error :-

* validate = [ status\_code : 422, message, string, errors : object]
* bad requset = [ status\_code : 400, message : string ]

Technician login api : -

url : host/technicians/login

method : POST

response type : JSON

body :

* password
* tech\_mobile

response :

On success :

Status\_code = 100

Data = [ access\_token : token, token\_type : bearer, expires\_in : int(0), status\_code : int(100)]

On error :

* Sending any data is wrong = [ status\_code : 104, error\_message : string ]
* Send password ron = [ status\_code : 102, error\_message: string ]

Technician profile api : -

url : host/technicians/profile

method : POST

response type : JSON

headers : {

Authorization : bearar toekn,

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, technician : techicina data ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

get packages api : -

url : host/technicians/get\_packages

method : GET

response type : JSON

headers : {

Authorization : bearar toekn,

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, packages : packages data ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

get models api : -

url : host/technicians/get\_models

method : GET

response type : JSON

headers : {

Authorization : bearar toekn,

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, models : models data ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

get subscriptions api : -

url : host/subscriptions/

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, subscriptions : subscriptions data ]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

change subscriptions status by id api : -

url : host/subscriptions/change\_status/sub\_id

method : PUT

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 103

Data = [ status\_code : int, message : string]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

subscription to new package api : -

url : host/subscriptions/sub\_package

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

get posts api : -

url : host/posts/

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, posts : posts data ]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

get comments by post id api : -

url : host/posts/comments/post\_id

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, comments : comments data ]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

add comment to post api : -

url : host/posts/add\_comment

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* Comment\_post (post\_id)
* Comment\_text
* Comment\_tech (tech\_id)

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, comments : comments data ]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

get view post by post id api : -

url : host/posts/post\_view/post\_id

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response : -

On success :

Status\_code = 100

Data = [ status\_code : int, message : string, post : post data ]

If it was successful but there is no data in the database = [ status\_code : 108, message: string ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

add new post api : -

url : host/posts/store\_post

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* title
* body
* photo

Response :

On success :

Status\_code = 100

Data = [ status\_code : int, message: string, post:post data ]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

add like to post by post id api : -

url : host/posts/like

method : POST

response type : JOSN

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* like\_tech ( tech\_id )
* like\_post ( post\_id)

Response :

On success :

Status\_code = 100

Data = [ status\_code : int, message: string]

On error :

Device token not sended = [ status : 106, error : string ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

view post api : -

url : host/posts/add\_post\_view

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* view\_device
* view\_tech (tech\_id)
* view\_post ( post\_id)

Response :

On success :

Status\_code = 100

Data = [ status\_code : int, message: string]

On error :

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

post cost api : -

url : host/posts/post\_cost

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response :

On success :

Status\_code = 100

Data = [ status\_code :int , messages: string ]

On error

Status\_code = 104 , technician posit = 0

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

create chat room api : -

url : host/chats/create\_room

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* room\_type
* room\_name

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , room : room data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

add member to chat room api : -

url : host/chats/add\_member

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* member\_tech ( tech\_id)
* member\_room ( room\_id)

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , member : member data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

Add new message api : -

url : host/chats/create\_message

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* msg\_from ( tech\_id)
* masg\_room (room\_id)
* msg\_context

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , message : message data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

get tickets api : -

url : host/tickets/

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , tickets : tickes data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

add new ticket api : -

url : host/tickets/add

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* ticket\_brand
* ticket\_model
* ticket\_category
* ticket\_tech
* ticket\_context

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , ticket : ticke data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

Status\_code = 107 ( not category)

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

add reply to ticket api : -

url : host/tickets/add\_ reply

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* reply\_ticket
* reply\_tech
* reply\_context

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , reply : reply data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

get ticket replies by ticket id api : -

url : host/tickets/replies/ticket\_id

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , replies : replies data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

get courses api : -

url : host/courses/

method: GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , courses : courses data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

Status\_code = 107 , ( no courses )

token not sended = [ status : 401, error : string ]

add view course api : -

url : host/courses/view

method : POST

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Body :

* course\_id

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , course : course data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]

validate = [ status\_code : 422, message, string, errors : object]

bad requset = [ status\_code : 400, message : string ]

get ads api : -

url : host/ads/

method : GET

response type : JSON

Authorization : bearar toekn,

headers : {

device\_Token : device\_Token

}

Response :

On success :

Status\_code = 100

Data = [ status\_code : int , message : string , ads : ads data ]

On error

Device token not sended = [ status : 106, error : string, view : view data ]

token not sended = [ status : 401, error : string ]