**API Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 (v1) | 05-Jun-2014 | Ryan | Initial draft |

**Index**

1. Authentication

1.1 mobile\_app\_authentication

Request

Response

1.2 mobile\_app\_logout

Request

Response

1.3 forgot\_password

Request

Response

2. Registration

2.1 sign\_up

Request

Response

2.2 update\_availability

Request

Response

2.3 get\_credit\_cards

Request

Response

2.4 add\_credit\_card

Request

Response

2.5 activate\_credit\_card

Request

Response

2.6 remove\_credit\_card

Request

Resposne

2.7 update\_bank\_details

Request

Response

2.8 get\_courses\_for\_subject

Request

Response

2.9 add\_courses

Request

Response

2.10 update\_profile

Request

Response

3. User

3.1 get\_profile

Request

Response

3.2 get\_reviews

Request

Response

3.3 get\_availability

Request

Response

3.4 get\_rate

Request

Response

3.5 get\_upcoming\_session

Request

Response

3.6 get\_session\_history

Request

Response

.3.7 get\_profile\_visits

Request

Response

3.8 get\_buddies

Request

Response

3.9 send\_message

Request

Response

3.10 get\_dashboard

Request

Response

3.11 get\_conversations

Request

Response

3.12 get\_messages

Request

Response

~~3.13 get\_events~~

~~Request~~

~~Response~~

3.14 Add\_friend

Request

Response

3.15 Remove\_friend

Request

Response

3.16.pusher\_auth

Request

Response

4. Group

4.1 get\_all\_groups

Request

Response

4.2 get\_group

Request

Response

4.3 get\_group\_members

Request

Response

5. Session

5.1 get\_session\_details

Request

Response

5.2 accept\_session

Request

Response

5.3 reject\_session

Request

Response

5.4 add\_review

Request

Response

5.5 edit\_review

Request

Response

5.6 get\_review\_details

Request

Response

5.7 book\_session

Request

Response

5.8 reschedule\_session

Request

Response

5.9 get\_invoice\_by\_session

Request

Response

5.10 get\_all\_invoices

Request

Response

5.11 pay\_for\_session

Request

Response

5.12 get\_availability\_for\_date

Request

Response

6. Search

6.1 search\_service\_provider

Request

Response

6.2 search\_courses\_or\_trades

Request

Response

6.3 search\_subject

Request

Response

6.4 global\_search

Request

Response

6.5 search\_students

Request

Response

6.6 search\_category

Request

Response

7.Universities

8. Tasks

8.1 get\_tasks

Request

Response

8.2 get\_task

Request

Response

8.3 add\_task

Request

Response

8.4 update\_task

request

response

8.5 remove\_Task

request

response

# 1. Authentication

 Authentication entity has actions that provides security to api. This api actions includes authentication token management ( auth\_token ). Login actions generate unique and secure auth\_token and provides it to mobile device. This token is used in communication between mobile device and server, where server use this token to identify user and allow access to api resources, only if valid user identified. Logout action in entity destroys auth\_token. Authentication api actions include following mobile api actions:

## 1.1 mobile\_app\_authentication

Authentication point of mobile app. It takes credentials and returns auth\_token.

### Request

|  |  |
| --- | --- |
| **Method**  | **URL**  |
| **POST** | mobile\_app\_api/v1/authentication/mobile\_app\_authentication  |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values**  |
| POST | email | string |
| POST | password | string  |

### Response

|  |  |
| --- | --- |
| **Status** | **Response**  |
| 200  | {  "status": 200, "auth\_token": "3a020793-70a7-4b79-ba63-1b067959aa77", "user\_info": { "id": 1475, "name": "Rakesh\_customer", "image": "users/original.png", "email": "rakesh\_customer@tutorscoop.com", "type": "Student" }}<auth\_token> (string) : unique authentication token used to validate and identify user at server side.e.g. e9322d4e-9816-4006-a936-0609096bd05b |
| 400  | { "status" : 400, "error" : "Login failed. Please try again with valid credentials."} |
| 500  | { "status": 500, "error" : "Something went wrong. Please try again with valid credentials."}  |
| 400 |  { status: 400 , error: "Kindly activate your account from your registered email address, using activation email sent to you." } |

## Or

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values**  |
| POST | provider | String (facebook/linkedin) |
| POST | uid | String (provider uid) |

### Response

|  |  |
| --- | --- |
| **Status** | **Response**  |
| 200  | {* **"status"**: 200,
* **"auth\_token"**: "b182f0ba-0019-4089-ac5f-c98280cfbf3a",
* **"user\_info"**: {
	+ **"id"**: 46,
	+ **"name"**: "Ramesh",
	+ **"image"**: "users/original.png",
	+ **"email"**: "ramesh.aconnexion@gmail.com",
	+ **"type"**: "Student",
	+ **"average\_rating"**: null,
	+ **"profile\_visits"**: 0,
	+ **"hourly\_rate\_in\_cents"**: null,
	+ **"university\_name"**: "Georgetown",
	+ **"school\_name"**: null,
	+ **"graduation\_year"**: null,
	+ **"major"**: null,
	+ **"minor"**: null

}}<auth\_token> (string) : unique authentication token used to validate and identify user at server side.e.g. e9322d4e-9816-4006-a936-0609096bd05b |
| 400  | { "status" : 400, "error" : "Login failed. Please try again with valid credentials."} |
| 500  | { "status": 500, "error" : "Something went wrong. Please try again with valid credentials."}  |
| 404 | {* **"status"**: 404,
* **"error"**: "User not registered."

} |

## 1.2 mobile\_app\_logout

This is a logout function of api. This api action deletes auth\_token from user record.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**   |
| **GET** | mobile\_app\_api/v1/authentication/mobile\_app\_logout/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values**  |
| GET | auth\_token | string  |

### Response

|  |  |
| --- | --- |
| **Status** | **Response**  |
| 200 | { "status": 200,"message": "Successfully Logged Out."} |
| 400  | { "status" : 400, "error" : "No user found with provided auth\_token."} |
| 500  | {  "status" : 500,  "error" : "Something went wrong. Please try again with valid credentials." } |

## 1.3 forgot\_password

This api action sends password reset instruction email to user.

### Request

|  |  |
| --- | --- |
| **Method**  | **URL**  |
| **GET** | mobile\_app\_api/v1/authentication/forgot\_password?email=<email> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params**  | **Values** |
| GET | email  | string |

### Response

|  |  |
| --- | --- |
| **Status** | **Response**  |
| 200  | { "status" : 200,"success" : ”Password reset instructions has been sent on <email> email address. Please check your mailbox.” } |
| 400  | { "status" : 400, "error" : "No user found with provided auth\_token."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again with valid credentials."}  |

# 2. Registration

 Registers users in system as a service provider or as a consumer. Registration process also generates auth\_token for user. This entity has following api actions:

## 2.1 sign\_up

This action registers user in system, generates unique auth\_token as well as returns this auth\_token as a response.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/registration/sign\_up |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | user | has |

user is a hash that holds user details. Example is as follow :

“user” =>

{

"name" => "Shibani",

"last\_name" => "Kashyup",

"email" => "\_sk@xyz.com",

"password" => "\*\*\*\*\*\*",

"phone" => "1234567890",

"type" => "Student / Tutor",

"password\_confirmation" => "\*\*\*\*\*\*",

"location" => NY,

"summary" => <small summary>,

"industry" => <industry to which user belong>,

}

Description of form elements:

1. user[name] : name of the user.

2. user[last\_name] : last name of the user.

3. user[email] : email of the user, must be well formed and unique.

4. user[password] : alphanumeric secret for user account.

5. user[password\_confirmation] : must be same as password.

6. user[type] : type of user, must be ‘Student’ (for consumer) or ‘Tutor’ (for service provider)

7. user[location] : location of user

8. user[summary] : small description of user

9. user[industry] : industry to which use belong

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200  | { "status" : 200,is\_valid\_user" : true,“user\_info” : <user\_record>,"auth\_token" : <auth\_token>} <auth\_token> (string) : unique authentication token for user.<user\_record> (json) : user database entity tuple, represented as json data which has structure as follow :{ id: integer,  name: string,  email: string,  crypted\_password: string,  salt: string,  activation\_state: string,  activation\_token: string,  activation\_token\_expires\_at: datetime,  reset\_password\_token: string,  reset\_password\_token\_expires\_at: datetime, reset\_password\_email\_sent\_at: datetime,  remember\_me\_token: string,  remember\_me\_token\_expires\_at: datetime,  created\_at: datetime, updated\_at: datetime,  type: string, admin: boolean,  school\_id: integer,  rate\_in\_cents: integer,  session\_duration: integer,  uuid: string,  market\_token\_id: string,  market\_refund\_token\_id: string,  market\_recipient\_email: string,  availability\_defaults: text,  min\_session\_length: integer,  max\_session\_length: integer,  avatar\_file\_name: string,  avatar\_content\_type: string,  avatar\_file\_size: integer,  avatar\_updated\_at: datetime,  about\_me: text,  billed\_length: integer,  discounts: text, response\_time: text,  education: text, header: text, has\_amazon\_account: boolean,  searchable: boolean,  mailing\_list: boolean,  last\_name: string, phone: string,  accepted\_compensation\_policy: boolean,  ip\_accepted\_compensation\_policy: string,  signature: string,  organization\_group: string,  card\_uri: string,  credits: integer,  bank\_uri: string,  balanced\_customer\_uri: string,  credits\_booked: integer,  points: integer,  streak: integer,  level: integer,  university: string, edu\_school: string,  major: string,  minor: string,  expected\_date\_of\_graduation: date,  gpa: decimal,  exp\_title: string,  exp\_location: string,  exp\_time\_period: string,  exp\_description: text,  hon\_title: string,  hon\_issuer: string,  hon\_date: date,  hon\_description: text, postal\_code: string,  home\_town: string,  additional\_needs: text,  minimum\_level: integer, balanced\_api\_id: text,  account\_no: string,  routing\_no: string,  is\_tutor\_onboard: integer,  is\_featured: boolean,  first\_login: boolean,  is\_tutor\_activated: integer,  is\_manual\_deactivated: integer,  billing\_address\_1: string,  billing\_address\_2: string,  billing\_city: string,  billing\_state: string,  billing\_zipcode: string,  shipping\_address\_1: string,  shipping\_address\_2: string,  shipping\_city: string,  shipping\_state: string,  shipping\_zipcode: string} |
| 400 | { "status" : 400, “error" : "Sign up process failed." "is\_valid\_user" : false, "error\_details" : <error\_details>} <error\_details>(array of string) : This string array contains results of server side data validation process. Format is as follow: [ “error 1”, "error 2",..., “error n” ] Error types : 1. "Name can't be blank" 2. "Email can't be blank" 3. "Email is not well formed" 4. "Password should match confirmation" 5. "Password can't be blank" 6. "Type is not included in the list" - occurs if type does not match ‘Student’ or ‘Tutor’ |
| 500  | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 2.2 update\_availability

Using update\_availability api action, availability of service provider can be changed. It takes availability data as stringified json object.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST**  | mobile\_app\_api/v1/registration/update\_availability/auth\_token |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST  | availability\_data | string |

availability\_data is a hash in string format.

Example :

{"availability": {

"0": [[21600,42300]],

"1": [[21600,42300]],

"2": [[21600,42300]],

"3": [[21600,42300]],

"4": [[21600,42300]],

"5": [[21600,42300]],

"6": [[21600,42300]]

}

}

In above example, keys represents days of week and values represent time in seconds from midnight 12.

|  |
| --- |
|  **Day** |
| 0 - Sunday |
| 1 - Monday |
| 2 - Tuesday |
| 3 - Wednesday |
| 4 - Thursday |
| 5 - Friday |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200  | { "status" : 200 } |
| 400 | { "status" : 400, "error" : “Can't update availability.”}  |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."}  |

##

## 2.3 get\_credit\_cards

Get credit card details of customer.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/registration/get\_credit\_cards/auth\_token |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "credit\_card\_details": [“Some credit card detail”]} |
| 400  | { "status": 400, "error":"Invalid user. Login and try again." } |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."}  |

## 2.4 add\_credit\_card

Add credit card details of customer.

### Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | mobile\_app\_api/v1/registration/add\_credit\_cards/auth\_token |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | credit\_card | hash |

credit\_card is a parameter hash with following structure:

"credit\_card"=> {

"number"=>"4111111111111111",

"expiry\_date"=>"04/2019",

"name"=>"Rakesh",

"cvv"=>"123"

}

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “success” : “Successfully created card.” } |
| 400 | { "status" : 400, “success” : “Invalid card details.” } |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 2.5 active\_credit\_card

Activated credit card of customer.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET**  | mobile\_app\_api/v1/registration/active\_credit\_card/auth\_token/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET  | id | Credit\_card ids |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "success": "Card successfully activated."} |
| 400 | {"status": 400,"error":"Invalid user. Login and try again." } |
| 500 | {"status": 500,"error": "Something went wrong. Please try again with valid credentials.","error\_details": "\"Couldn't find CreditCard with 'id'=126 [WHERE `credit\_cards`.`user\_id` = ?]\""} |

## 2.6 remove\_credit\_card

Add credit card details of customer.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET**  | mobile\_app\_api/v1/registration/remove\_credit\_card/auth\_token/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | id | Credit\_card ids |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "success": "Successfully destroyed card."} |
| 400  | { "status": 400, "error": "Can't destroy card."} |
| 500 | {"status": 500,"error": "Something went wrong. Please try again with valid credentials.","error\_details": "\"Couldn't find CreditCard with 'id'=126 [WHERE `credit\_cards`.`user\_id` = ?]\""}  |

## 2.7 update\_bank\_details

Creates or updates bank account details of service provider.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/registration/update\_bank\_details |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | payment | hash |

payment is a parameter hash with following structure:

"payment" =>

 {

 "name"=>"Kiran Sarode",

 "bank\_code"=>"021000021",

 "account\_number"=>"9900000002",

 "type"=>"SAVINGS"

 }

###

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “success” : “Successfully created bank account.” } |
| 400 | { "status" : 400, “success” : “Invalid bank details.” } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 2.8 get\_courses\_for\_subject

This api action returns all courses belonging to subject with id provided.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/registration/get\_courses\_for\_subject |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | subject\_id | integer |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  “courses” : <courses\_details> “total\_courses” : <courses\_count> }<courses\_details> (json array) : user database entity tuple, represented as json data which has structure as follow :{ "created\_at": <creation date and time>, "description": <description>, "id": <course id>, "name": <course name>, "number": null, "school\_id": <school id to which course belong>, "subject\_id": <subject id>, "updated\_at": <last update date and time>, "\_type": "course"}<courses\_count> (integer) : count of total number of courses returned. |
| 400 | {  "status" : 400,  “success” : “Subject with subject\_id <subject\_id> not found.” } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 2.9 add\_courses

This api action adds courses to tutors profile.

###

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/registration/add\_courses |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | selected\_course\_ids | array of course ids |
| POST | subject\_id | integer |

selected\_course\_ids is an array of course ids.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200 } |
| 400 | { "status" : 400, "error" : “Course with id <course>not found.”} |
| 400 | {  "status" : 400,  "error" : “Array of course ids not found in parameters, please check course id array and try again.”} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

##

## 2.10 update\_profile

This api action updates service provider’s and consumer’s profile information.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/registration/update\_profile |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | student / tutor | hash |

student and tutor are hashes which contains user information. Structure of parameter object is different for both type of user i.e service provider and consumer.

For consumers (student / customer) :

"student"=>{

"name"=> <name of student / customer>,

 "school\_id"=> <name of school>,

 "major"=> <major field>,

 "phone"=> <phone number>,

 "organization\_group"=> <organization name>

 }

For service providers (tutor / fixxpert) :

 "tutor"=>{

 "name"=> <name of tutor / fixxpert>,

 "last\_name"=> <last name of tutor / fixxpert>,

"location"=> <location of tutor / fixxpert>,

"rate"=> <rate of tutor / fixxpert>,

"summary"=> <summary of tutor / fixxpert>,

 "exp\_title"=> <subtile or experience title>,

 "school\_id"=> <name of school>,

 "major"=> <major field>,

 "rate"=> <rate in dollor>,

 "response\_time"=> <response time>,

 "about\_me"=> <more details / biography>,

 "industry"=> <industry to which use belong>,

 "headline"=> <position>,

}

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200 } |
| 400 | { "status" : 400, "error" : “Update failed”} |
| 400 | { "status" : 400, "error" : “User with auth\_token <autn\_token> not found.” } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

##

# 3. User

GuideU and Fixxpert has a entity called user which holds all user information. User information is sub classified as tutor / fixxpert and student / customer. Following are the api actions related to user entity.

## 3.1 get\_profile

This api action returns complete user data. it fetches tuple from database and returns in the form of json object.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_profile/<auth\_token>?user\_id=<user\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | user\_id | integer (Optional) |

user\_id is an id of user and it is OPTIONAL. If user\_id parameter is present in request, it will return information of user whose id provided else it will return information of user who has logged in and is identified by auth\_token.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “user\_info” : { <user\_record>, “user\_courses”: <array of courses user is taking, along with object of SECTION for that course>, “user\_skills”: <array of skills user is taking>, “user\_organizations”: <array of organizations subscribed to>, “user\_sports”: <array of user\_sports selected>, If User is a Tutor “tutor\_courses”: <array of courses tutor can teach>, “tutor\_skills”: <array of skills tutor can teach> } “conversation\_id” : <conversation id with given user\_id> “reviews” : < recieved\_reviews >}<user\_record> (json) : user database entity tuple, represented as json data which has structure as follow :{"status":200,"user\_info":{"id":31,"name":"test50","email":"test50@georgetown.edu","activation\_state":"active","activation\_token":null,"activation\_token\_expires\_at":null,"reset\_password\_token":null,"reset\_password\_token\_expires\_at":null,"reset\_password\_email\_sent\_at":null,"remember\_me\_token":null,"remember\_me\_token\_expires\_at":null,"created\_at":"2015-01-22T04:31:09.000-05:00","updated\_at":"2015-01-23T00:08:57.000-05:00","type":"Tutor","admin":false,"school\_id":null,"rate\_in\_cents":2000,"session\_duration":null,"market\_token\_id":null,"market\_refund\_token\_id":null,"market\_recipient\_email":null,"availability\_defaults":null,"min\_session\_length":null,"max\_session\_length":null,"billed\_length":null,"discounts":null,"response\_time":null,"education":null,"header":null,"has\_amazon\_account":false,"searchable":false,"mailing\_list":false,"last\_name":"ramesh","phone":"3432423424","accepted\_compensation\_policy":false,"ip\_accepted\_compensation\_policy":null,"signature":null,"organization\_group":null,"credits":0,"balanced\_customer\_uri":null,"credits\_booked":0,"points":0,"streak":0,"level":1,"university":null,"edu\_school":null,"expected\_date\_of\_graduation":null,"gpa":null,"exp\_title":"MyOneLineDescription","exp\_location":null,"exp\_time\_period":null,"exp\_description":null,"hon\_title":null,"hon\_issuer":null,"hon\_date":null,"hon\_description":null,"postal\_code":null,"home\_town":null,"additional\_needs":null,"minimum\_level":1,"balanced\_api\_id":null,"account\_no":null,"routing\_no":null,"is\_tutor\_onboard":null,"is\_featured":false,"first\_login":false,"is\_registration\_completed":true,"is\_manually\_deactivated":false,"billing\_address\_1":null,"billing\_address\_2":null,"billing\_city":null,"billing\_state":null,"billing\_zipcode":null,"shipping\_address\_1":null,"shipping\_address\_2":null,"shipping\_city":null,"shipping\_state":null,"shipping\_zipcode":null,"location":null,"experience":null,"summary":null,"awards":null,"skills":null,"industry":null,"provider":null,"uid":null,"profile\_step":6,"headline":null,"onboard\_step":null,"pending\_chat\_with":null,"cached\_notification\_count":0,"cached\_message\_count":0,"university\_id":2,"level\_changed":null,"skills\_count":1,"courses\_count":2,"image":"users/original.png","user\_courses":[{"id":23345,"school\_id":null,"subject\_id":396,"name":"ACCT 500","number":null,"description":null,"created\_at":"2015-01-13T04:36:16.000-05:00","updated\_at":"2015-01-13T04:36:16.000-05:00","university\_id":2,"\_type":"course","section":{"id":41254,"course\_id":23345,"name":"01","university\_id":2,"created\_at":"2015-01-13T04:36:16.000-05:00","updated\_at":"2015-01-13T04:36:16.000-05:00"}},{"id":14472,"school\_id":null,"subject\_id":424,"name":"EMPL 844","number":null,"description":null,"created\_at":"2015-01-13T02:43:59.000-05:00","updated\_at":"2015-01-13T02:43:59.000-05:00","university\_id":2,"\_type":"course","section":{"id":29259,"course\_id":14472,"name":"30","university\_id":2,"created\_at":"2015-01-13T02:43:59.000-05:00","updated\_at":"2015-01-13T02:43:59.000-05:00"}}],"tutor\_courses":[{"id":23345,"school\_id":null,"subject\_id":396,"name":"ACCT 500","number":null,"description":null,"created\_at":"2015-01-13T04:36:16.000-05:00","updated\_at":"2015-01-13T04:36:16.000-05:00","university\_id":2,"\_type":"course"},{"id":23338,"school\_id":null,"subject\_id":396,"name":"ACCT 001","number":null,"description":null,"created\_at":"2015-01-13T04:36:07.000-05:00","updated\_at":"2015-01-13T04:36:07.000-05:00","university\_id":2,"\_type":"course"}],"user\_skills":[{"id":35,"name":"Ross Essays","created\_at":"2015-01-22T06:53:06.000-05:00","updated\_at":"2015-01-22T06:53:06.000-05:00","university\_id":2,"\_type":"skill"}],"tutor\_skills":[{"id":35,"name":"Ross Essays","created\_at":"2015-01-22T06:53:06.000-05:00","updated\_at":"2015-01-22T06:53:06.000-05:00","university\_id":2,"\_type":"skill"}],"user\_organizations":[{"id":336,"name":"Academic Resource Center (ARC)","university\_id":2,"created\_at":"2015-01-22T06:48:07.000-05:00","updated\_at":"2015-01-22T06:48:07.000-05:00"}],"user\_sports":[{"id":1,"name":"MENS BASEBALL","university\_id":2,"created\_at":"2015-01-22T06:49:38.000-05:00","updated\_at":"2015-01-22T06:49:38.000-05:00"}],"average\_rating":0.0,"profile\_visits":0,"hourly\_rate\_in\_cents":2500,"university\_name":"Georgetown","school\_name":"MySchool","graduation\_year":2015,"major":"MyMajor","minor":"MyMinor"},"conversation\_id":null,"reviews":[],"is\_friend":false} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.2 get\_reviews

get\_review returns all reviews given to service provider by his/her student / customer.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_reviews/<auth\_token>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “reviews” : <reviews\_details>}<reviews\_details> (json array) : json array which holds comments details and has following structure :{ average\_review: "4.5", clarity: 4, course\_id: 54, created\_at: "2014-02-15T00:07:15-06:00", easiness: 4, for\_user\_id: 2; helpfulness: 4, id: 5, professionalism: 5, rater\_interest: 5, text: "Great session this time... ", updated\_at: "2014-02-23T05:19:16-06:00", user\_id: 9, from\_user:  {name: "Jay D Homse",image: <image path in string format>,type: "Student" }} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.3 get\_availability

This api action returns weekly availability status of service provider.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_availability/auth\_token/tutor\_id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “availability” : <availability\_details>}<availability\_details> (json array) : json array which holds comments details and has following structure :{ 0: ["28800","29700"], 1: ["28800"], 2: ["28800"], 3: ["28800"], 4: ["28800"], 5: ["28800"], 6: ["28800"]}Here, Keys 0,1,2,3… represents days of week i.e. sunday, monday, tuesday… Numbers are seconds from midnight 12 O'clock. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.4 get\_rate

This api action returns hourly rate of service provider and currency information.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_rate/<auth\_token>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “rate” : <rate\_details>}<rate\_details> (json) : returns rate per session in cents, also contains currency details. Response structure is as follow :{ fractional: "5500.0", currency:  {id: "usd",priority: 1,iso\_code: "USD",name: "United States Dollar",symbol: "$",alternate\_symbols: ["US$"],subunit: "Cent",subunit\_to\_unit: 100,symbol\_first: true,html\_entity: "$",decimal\_mark: ".",thousands\_separator: ",",iso\_numeric: "840" }, bank:  {rounding\_method: null,rates: { },mutex: { } }} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.5 get\_upcoming\_session

get upcoming\_session action returns details of all upcoming confirmed sessions.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_upcoming\_session/<auth\_token>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “upcoming” : <upcoming\_session\_details>}<upcoming\_session\_details> (json) : returns details for upcoming sessions. Structure is as follow :{ id: integer (session id),  tutor\_id: integer (id of tutor/fixxpert),  student\_id: integer (id of student/customer), date: date (date of session),  start\_time: integer(start time in seconds from midnight 12),  length: integer(duration of sessions in minutes),  created\_at: datetime,  updated\_at: datetime,  status: string (confirm - c / rejected - c),  length\_out\_schedule: integer,  student\_accepted: boolean,  occured: boolean,  end\_time: datetime,  course: string,  topic: string,  location: string,  course\_id: integer (id of course / trade),  created\_by\_type: string} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.6 get\_session\_history

This api action returns history of sessions i.e. past confirmed sessions.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_session\_history/<auth\_token>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “previous” : <past\_session\_details>}<past\_session\_details> (json) : returns details for previous/past sessions. Structure is as follow :{ id: integer (session id),  tutor\_id: integer (id of tutor/fixxpert),  student\_id: integer (id of student/customer), date: date (date of session),  start\_time: integer(start time in seconds from midnight 12),  length: integer(duration of sessions in minutes),  created\_at: datetime,  updated\_at: datetime,  status: string (confirm - c / rejected - c),  length\_out\_schedule: integer,  student\_accepted: boolean,  occured: boolean,  end\_time: datetime,  course: string,  topic: string,  location: string,  course\_id: integer (id of course / trade),  created\_by\_type: string} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.7 get\_profile\_visits

It returns the count of total number of users who visited profile of particular service provider. It also provides the details of visits by each user.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_profile\_visits/<auth\_token>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |

tutor\_id is an id of Tutor / Fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “visit\_count” : <visit\_count>, “visits” : <details\_of\_visits>}<visit\_count> (integer) : count of total users visited tutor’s / fixxpert’s profile. <details\_of\_visits> (array of json) : Structure is as follow :{ id: integer (visit id),  student\_id: integer (visitor id i.e. student’s / customer’s id),  count: integer (visitor’s visit count),  created\_at: datetime,  updated\_at: datetime,  user\_id: integer ( tutor’s / fixxpert’s id whos profile has been visited )} |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "No service provider found with id <tutor\_id>."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.8 get\_buddies

This action returns buddies (users with interaction) to display in chatting buddy list.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_buddies/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “buddies” : <buddies\_array>, “buddy\_count” : <count\_of\_buddies>}<count\_of\_buddies> (integer) : count of buddies returned in response.<buddies\_array> (array of json) : array of json objects which contains buddy details and each json object has structure similar to the structure of object that get\_profile API action returns. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.9send\_message

This api action receives sender\_id, receiver\_id, message, processes it and pushes processed contents on appropriate channel. send\_message api can be used for both one to one chat and for group chat / message broadcasting.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/user/send\_message |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | chat\_info | hash |

chat\_info is a hash which contains all necessary information for getting sender and receiver information and routing message to proper communication channel. Structure is as follow :

chat\_info = {

to : <recipient id>, (This can be a user id or group id)

from : <sender id>,

text : <actual message to be sent>,

is\_broadcast : 0 for no / 1 for yes

}

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200 } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : “User with auth\_token <auth\_token> not found.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.10 get\_dashboard

This action identifies user from auth\_token and return information necessary for dashboard. Response object is different for service providers as well as for customers.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_dashboard/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200, "sessions" : <session\_details>, "related\_users" : <user\_details>, “conversation\_ids”: <conversation\_ids>, “events”: <events>, “group\_events” : <group\_event>, “ groups” : <groups>, “ group\_members” : <user\_info>, “user\_courses”: <courses>, “user\_skills” : <skills>, “user\_organizations”: <organizations> “user\_sports”: <sports>} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later." } |

Example: -

{

* **"status"**: 200,
* **"sessions"**: {
	+ **"upcoming"**: {
		- **"as\_customer"**: [ ],
		- **"as\_fixxpert"**: [ ]

},

* + **"previous"**: {
		- **"as\_customer"**: [ ],
		- **"as\_fixxpert"**: [ ]

},

* + **"pending"**: {
		- **"as\_customer"**: [ ],
		- **"as\_fixxpert"**: [ ]

}

* },
* **"related\_users"**: { },
* **"conversation\_ids"**: [ ],
* **"user\_events"**: [ ],
* **"group\_events"**: [
	+ {
		- **"id"**: 7,
		- **"resource\_id"**: 6618,
		- **"resource\_type"**: "Group",
		- **"name"**: "asd",
		- **"description"**: "asd",
		- **"start\_time"**: "2015-01-16T08:00:00.000-05:00",
		- **"end\_time"**: "2015-01-16T11:00:00.000-05:00",
		- **"status"**: "Pending",
		- **"frequency"**: "Daily"

},

* + {
		- **"id"**: 8,
		- **"resource\_id"**: 6619,
		- **"resource\_type"**: "Group",
		- **"name"**: "sad",
		- **"description"**: "sad",
		- **"start\_time"**: "2015-01-20T17:15:00.000-05:00",
		- **"end\_time"**: "2015-01-20T20:30:00.000-05:00",
		- **"status"**: "Pending",
		- **"frequency"**: "Weekly"

}

* ]

}

## 3.11 get\_conversations

This action returns conversations (users with interaction) to display in chat conversation

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_conversations/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | page | Integer |
| GET | Per\_page (default=20)  | Integer  |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “conversations” : <conversations\_array>}<conversations\_array> (array of json) : array of json objects which contains chat conversations of that user. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : “No chat conversation found.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 3.12 get\_messages

This action returns messages

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/user/get\_messages/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | page | Integer |
| GET | Per\_page(default=20) | Integer  |
| GET | conversation\_id | integer |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “messages” : <messages\_array>}<messagess\_array> (array of json) : array of json objects which contains chat conversations's messages of that user for that conversation. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : “No Messages found.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

**~~3.13 get\_events~~**

~~This action returns events between specific dates~~

### ~~Request~~

|  |  |
| --- | --- |
| **~~Method~~** | **~~URL~~**  |
| **~~GET~~** | ~~mobile\_app\_api/v1/user/get\_events/<auth\_token>~~ |

|  |  |  |
| --- | --- | --- |
| **~~Type~~** | **~~Params~~** | **~~Values~~** |
| ~~GET~~ | ~~start\_date~~ | ~~e.g. '2015-01-01'~~ |
| ~~GET~~ | ~~end\_date~~ | ~~e.g. '2015-12-12'~~ |

### ~~Response~~

|  |  |
| --- | --- |
| **~~Status~~** | **~~Response~~** |
| ~~200~~ | ~~{~~ ~~"status" : 200,~~ ~~“events” : <events>~~~~}~~~~<events\_array> (array of json) : array of json objects which contains events of that user for given dates.~~ |
| ~~401~~ | ~~{ "status" : 401, "error" : “Invalid user. Login and try again.”}~~ |
| ~~400~~ | ~~{ "status" : 400, "error" : “No Events found.”}~~ |
| ~~500~~ | ~~{ "status" : 500, "error" : "Something went wrong. Please try again later."}~~ |

**3.14 Add\_friend**

This action add user to friend's list.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/user/add\_friend/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | user\_id | <user\_id> |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “success” : <success\_message>} |
| 401 | { "status" : 401, "notice" : “<user> is already your friend.”} |
| 400 | { "status" : 400, "error" : “Please provide valid user id.”} |

**3.15 Remove\_friend**

This action remove user from friend's list.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/user/remove\_friend/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | user\_id | <user\_id> |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “success” : <success\_message>} |
| 401 | { "status" : 401, "notice" : “<user>is not in your friend's list.”} |
| 400 | { "status" : 400, "error" : “Please provide valid user id.”} |

**3.16 pusher\_auth**

This action will provide pusher key..

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/user/pusher\_auth/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | type | <type> |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “success” : <success\_message>} |
| 403 | { "status" : 401, "text" : Forbidden.”} |

# 4. Group

GuideU and Fixxpert has a group for each course, skills, organizations,intrests. Any user can join group as well as leave group. Groups are the best medium for topic discussion which is possible with message broadcasting feature.

## 4.1 get\_all\_groups

Returns all groups present in system.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/group/get\_all\_groups/<auth\_token>/<page> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | page | integer |

page is a pagination offset. Each page returns details of 5 groups.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “groups” : <group\_details> “tutors” :<tutors> “students”: <students> “friends”: <friends>}<page> (integer) : page offset sent in parameter.<groups> (array) : array of json data which contains group details <tutors> (array) : array of json data which contains tutors details <students> (array) : array of json data which contains students details <friends> (array) : array of json data which contains friends details  |
| 400 | { "status" : 400, "error" : No Group Present.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 4.2 get\_group

Returns all groups matching group id.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/group/get\_group/<auth\_token>/<qroup\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | Group\_id | integer |

query is a group\_id.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “groups” : <group\_details> “tutors” :<tutors> “students”: <students> “friends”: <friends>}<page> (integer) : page offset sent in parameter.<groups> (array) : array of json data which contains group details <tutors> (array) : array of json data which contains tutors details <students> (array) : array of json data which contains students details <friends> (array) : array of json data which contains friends details  |
| 400 | { "status" : 400, "error" : "Please provide valid group id."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 4.3 get\_group\_members

Returns all group members of specified group.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/group/get\_group\_members/<auth\_token>/<group\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | group\_id | integer |
| GET | type | Tutors/students/friends |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, “group\_members” : <group\_members>}<group\_id> (string) : group id sent in parameter.<group\_members>: Group member of given group id |
| 400 | { "status" : 400, "error" : "No group found with id <group\_id>."} |
| 400 | { "status" : 400, "error" : "Group ID provided is either null or empty. Try again with valid group id."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

# 5. Session

 Session is a period in which, service provider (tutor / fixxpert) serves consumers (student / customer). Users can book session, accept session, reject session or reschedule session. Session entity of api has following action:

## 5.1 get\_session\_details

Returns details of session.

###  Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/get\_session\_details/<auth\_token>/<session\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | session\_id | integer |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "session\_details" : <session\_details>, “service\_provider” : <service\_provider\_details>, “consumer” : <consumer\_details>} <session\_details> (json) : session details in json format, it has following structure: { course:  {created\_at: "2014-02-12T08:42:48-06:00",description: <session description>,id: <id of course / trade>,name: <name of course / trade>,number: null,school\_id: <id of school to which course belong>,subject\_id: <id of subject to which course / trade belong >,updated\_at: "2014-02-12T08:42:48-06:00",\_type: "course" }, course\_id: <id of course / trade>, created\_at: "2014-02-14T11:47:04-06:00", created\_by\_type: "Student", date: "2014-02-20", end\_time: "2014-02-20T12:00:00-06:00", id: 6, length: <duration of session in minutes>, length\_out\_schedule: 0, location: <location of session - string>, occured: null, start\_time:<start time in seconds from midnight 12>, status: <status of session - string - c:confirm, r:reject>, student\_accepted: false, student\_id: <consumer id>, topic: <topic of session - string>, tutor\_id: <service provider id>, updated\_at: "2014-02-14T11:47:45-06:00"}<service\_provider\_details> (json) : JSON object (database tuple is represented as json object) consists of complete tutor / fixxpert data. Refer database schema to obtain columns as key for json object.<consumer\_details> (json) : JSON object (database tuple isrepresented as json object) consists of complete user data. Refer database schema to obtain columns as key for json object. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "Session with id <session\_id> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.2 accept\_session

This api action accepts session with given session\_id.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/accept\_session/<auth\_token>/<session\_id>/<message> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | session\_id | integer |
| GET | message | string |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "success" : "Session confirmed." } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "Session with id <session\_id> not found."} |
| 400 | {  "status" : 400,  "error" : "Can't confirm session. Time slot is not available, please contact service provider."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.3 reject\_session

Rejects session with id provided in parameter as.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/reject\_session/<auth\_token>/<session\_id>/<message> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | session\_id | integer |
| GET | message | string |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "success" : "Session rejected." } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : "Session with id <session\_id> not found."} |
| 400 | {  "status" : 400,  "error" : "Can't reject session - This session is already rejected."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.4 add\_review

This action add review for service provider.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/session/add\_review/<auth\_token>/session\_id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | session\_id | integer |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "review\_id" : <id\_of\_new\_review>,  "success" : "Review created for <service\_provider> by <user>." } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : "<user\_name> can’t create review for <service\_provider>."} |
| 400 | {  "status" : 400,  "error" : "Service provider with id <tutor\_id>not found.""} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.5 edit\_review

Review can be updated using this api action.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/session/edit\_review |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "success" : "Review updated for <service\_provider> by <user>." } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : "<user\_name> can’t update review for <service\_provider>."} |
| 400 | {  "status" : 400,  "error" : "Service provider with id <tutor\_id> not found.""} |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.6 get\_review\_details

This api action provides review details such as helpfulness, interest, clarity, professionalism rates and user comments.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/get\_review\_details/<auth\_token>/<review\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | review\_id | integer |

<review\_id> : id of review

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "review" : <review\_details>}<review\_details> (json): Json object which holds review details with following structure : { "average\_review": <average review ratings> "clarity": <clarity rating (1 to 5) - integer>, "course\_id": <id of course - integer>, "created\_at": <creation date and time of record>, "for\_user\_id": <service provider id - integer>, "helpfulness": <helpfulness rating (1 to 5) - integer>, "id": 12, "professionalism": <professionalism rating (1 to 5) - integer>, "rater\_interest": <interest rating (1 to 5) - integer>, "text": <comment by consumer>, "updated\_at": <last update date and time>, "user\_id": <consumer id - integer> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : "Review with id <review\_id> not found.""} |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.7 book\_session

Session can be booked by using this api action with specified service provided.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/session/book\_session |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | tutor\_id | integer |
| POST | tutoring\_session | json |

<tutor\_id> : id of tutor / fixxpert

<tutoring\_session> : json object which holds details of session booking, which has structure as follow :

tutoring\_session" =>

{

"date" => <date of session - string (yyyy-mm-dd)>,

"start\_time" => <start time of session - string (HH:MM AM/PM)>,

"length" => <length in minutes>, OR end\_time either is fine with us.

resource\_id => <id of eiher course of skill – integer>,

resource\_type => <type of resource, string, 'Course', 'Skill'>

"on\_site" => <Boolean for session location>,

"location" => <location for session>,

"message" => <message for service provider>,

"user\_id" => <customer’s id>,

"tutor\_id" => <service provider’s id>

}

e.g. json ---curl -X POST http://localhost:3000/mobile\_app\_api/v1/session/book\_session/311cb2a9-1238-4c1f-af1f-82f2b6fa4025 -d "tutor\_id=3&tutoring\_session%5Bcategory\_id%5D=1&tutoring\_session%5Bcourse\_id%5D=1&tutoring\_session%5Bdate%5D=2014-09-18&tutoring\_session%5Bend\_time%5D=03%3A00&tutoring\_session%5Blocation%5D=dsa&tutoring\_session%5Bmessage%5D=nil&tutoring\_session%5Bon\_site%5D=false&tutoring\_session%5Bstart\_time%5D=02%3A45&tutoring\_session%5Bsubject\_id%5D=1&tutoring\_session%5Btutor\_id%5D=3&tutoring\_session%5Buser\_id%5D=2"

request -- curl -X POST http://localhost:3000/mobile\_app\_api/v1/session/book\_session/14ba7701-b6d8-413a-8df9-1e197134f6fa -d tutoring\_session: {category\_id: '1',subject\_id: '1',course\_id:'1',location:'dsa',start\_time:'02:45',end\_time: '03:00',message:'nil',tutor\_id:'3',user\_id:'2',on\_site:false,date:'2014-09-18'}, tutor\_id:3

outsput-------- {"status":200,"session\_details":{"id":10,"tutor\_id":3,"user\_id":2,"date":"2014-09-18","created\_at":"2014-09-17T09:12:26.337-04:00","updated\_at":"2014-09-17T09:12:26.337-04:00","status":null,"student\_accepted":false,"occured":null,"end\_time":"2014-09-17T17:30:00.000-04:00","location":"dsa","course\_id":1,"category\_id":1,"rate\_in\_cents":4000,"on\_site":false,"start\_time":"2014-09-17T17:15:00.000-04:00","subject\_id":"1","old\_session":false}}

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "session\_details" : <session\_details>}<session\_details> (json): Json object which holds session details with following structure : { "what": <course / trade name>, "where": <location for session>, "when": <date & time (MMM DD, YYYY HH:MM AM/PM)>, "notes" : <Message while booking session>, "tutor\_image\_url" : <image url>, "tutor\_name" : <tutor / fixxpert name>, "tutor\_rate" : <tutor / fixxpert rate>, "tutor\_email" : <tutor / fixxpert email> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : ”Session booking failed.”, "error\_deatils" : <error\_details>}<error\_details> (array of string) : Array of errors in string firmat.  [ “error 1”, "error 2",..., “error n” ]Error types : 1. "Consumer has no bank account."2. "Service provider is not available for this session." |
| 400 | {  "status" : 400,  "error" : "Service provider with id <tutor\_id> not found.""} |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.8 reschedule\_session

Service provider can reschedule session. This can be done by using this api action.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/session/reschedule\_session |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | auth\_token | string |
| POST | session\_id | integer |
| POST | tutoring\_session | json |

<session\_id> : id of session to be rescheduled.

<tutoring\_session> : json object which holds details of session booking, which has structure as follow :

"tutoring\_session" =>

{

"date" => <date of session - string (yyyy-mm-dd)>,

"start\_time" => <start time of session - string (HH:MM AM/PM)>,

“resource\_id” => <id of course or Skill – integer>

“resource\_type” => <type of resource, 'Course' or 'Skill' - string>

"length" => <length in minutes>,

"location" => <location for session>

}

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "session\_details" : <session\_details>}<session\_details> (json): Json object which holds session details with following structure : { "what": <course / trade name>, "where": <location for session>, "when": <date & time (MMM DD, YYYY HH:MM AM/PM)>, "notes" : <Message while booking session>, "tutor\_image\_url" : <image url>, "tutor\_name" : <tutor / fixxpert name>, "tutor\_rate" : <tutor / fixxpert rate>, "tutor\_email" : <tutor / fixxpert email> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : ”Rescheduling failed.”, "error\_deatils" : <error\_details>}<error\_details> (array of string) : Array of errors in string format.  [ “error 1”, "error 2",..., “error n” ] |
| 400 | {  "status" : 400,  "error" : "Service provider with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

##

## 5.9 get\_invoice\_by\_session

This api action returns generated invoice for session.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/get\_invoice\_by\_session/<auth\_token>/<session\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | session\_id | integer |

<session\_id> : id of session of which invoice needed.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "invoice\_details" : <invoice\_details>}<invoice\_details> (json): Json object which holds invoice details with following structure : { "created\_at": <creation date and time>, "email\_notifications":<payment notification count - integer>, "id":<invoice id>, "last\_notification": <last notification date and time>, "manually\_billed": false, "paid":<payment status - true/false>, "payable\_amount":<payable amout in cents>, "rate\_in\_cents":<rate of session in cents>, "session\_length":<length of session in minutes>, "student\_id": <consumer id>, "tutor\_id": <id of service provider>, "tutoring\_session\_id": <id of session>, "updated\_at": <last update date and time> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | { "status" : 400, "error" : ”No invoice found for session\_id <session\_id>” } |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.10 get\_all\_invoices

This api action returns all invoices associated with a user.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/get\_all\_invoices/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |

<session\_id> : id of session to be rescheduled.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "invoices" : <invoice\_info\_arr>}<invoice\_info\_arr> (json array): Array of json object which holds invoice details with following structure : { "created\_at": <creation date and time>, "email\_notifications":<payment notification count - integer>, "id":<invoice id>, "last\_notification": <last notification date and time>, "manually\_billed": false, "paid":<payment status - true/false>, "payable\_amount":<payable amout in cents>, "rate\_in\_cents":<rate of session in cents>, "session\_length":<length of session in minutes>, "student\_id": <consumer id>, "tutor\_id": <id of service provider>, "tutoring\_session\_id": <id of session>, "updated\_at": <last update date and time> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

## 5.11 pay\_for\_session

This is a payment action by which user can pay fees of booked session.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/pay\_for\_session/<auth\_token>/<session\_id>/<tutor\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | tutor\_id | integer |
| GET | session\_id | integer |

<session\_id> : id of session, <tutor\_id> : id of service provider

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  "invoice" : <paid\_invoice\_details>, "success" : “Tutor is paid for session with id <session\_id>.”}<paid\_invoice\_details> (json): Json object which holds invoice details with following structure : { "created\_at": <creation date and time>, "email\_notifications":<payment notification count - integer>, "id":<invoice id>, "last\_notification": <last notification date and time>, "manually\_billed": false, "paid":<payment status - true/false>, "payable\_amount":<payable amout in cents>, "rate\_in\_cents":<rate of session in cents>, "session\_length":<length of session in minutes>, "student\_id": <consumer id>, "tutor\_id": <id of service provider>, "tutoring\_session\_id": <id of session>, "updated\_at": <last update date and time> } |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 400 | {  "status" : 400,  "error" : "No invoice found for session\_id <session\_id>."} |
| 400 | {  "status" : 400,  "error" : "User with auth\_token <auth\_token> not found."} |
| 400 | { “status” : 400, “error” : “Payment for session with id <session\_id> is failed. Its either already paid or something went wrong while processing payment operation.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

5.12 get\_availability\_for\_date

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/session/ get\_availability\_for\_date /<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | Start\_date | date |
| GET | tutor\_id | integer |
| GET | End\_date | date |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "status" : 200,  " availabilities " : < availabilities >, |
| 400 | {  "status" : 400,  "error" : " Session is not available”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again."} |

# 6. Search

GuideU and Fixxpert web applications has global search engine which provides the list of courses - subjects / trades and tutor / fixxpert in response to search query.

Following are the api actions related to search entity.

## 6.1 search\_service\_provider

This api action searches and returns service providers for given search query.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/search\_service\_provider/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it is a name of tutor / fixxpert.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <tutor/fixxpert\_count>, “service\_providers” : <tutor/fixxpert\_details>}<tutor/fixxpert\_details> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of complete tutor / fixxpert data. Refer database schema to obtain columns as key for json object.<tutor/fixxpert\_count> (integer) : Count of total tutor / fixxperts returned for <query>. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 6.2 search\_courses

This api action searches and returns courses / trades for provided search query.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/search\_courses\_or\_trades/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it is a name of course / trade.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <course\_count>, “courses\_or\_trades” : <course\_details>}<course\_details> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of complete course/trade data. Refer database schema to obtain columns as key for json object.<course\_count> (integer) : Count of total courses returned for <query>. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 6.3 search\_subject

This api action searches subjects / trades for provided query.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/search\_subject/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it can be a name of subject.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <subject\_count>, “subjects” : <subject\_details>}<subject\_details> (json array) : This is an array of json object. Each object provides details of subject and has structure as follow : { "created\_at": <creation date and time>, "description": <subject description>, "id":<subject id>, "is\_featured": <featured flag true/false>, "name": <name of subject>, "updated\_at": <last update date and time>, "\_type": "subject"}<subject\_count> (integer) : Count of total search result. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 6.4 global\_search

This actions returns all service providers and courses / trades (services) matching search query provided.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/global\_search/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it can be a name of tutor / fixxpert or a course / trade.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <course\_count>, “categories” : <categories>, “service\_providers” : <service\_providers\_details>, subjects" : <subjects>, “courses” : <courses>, “students” : <students >}<courses>/<categories> /<subjects> /<students> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of complete categories/subjects/courses/students data. Refer database schema to obtain columns as key for json object.<service\_providers\_details> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of complete tutor / fixxpert data. Refer database schema to obtain columns as key for json object.<search\_count> (integer) : Count of total search result. |

## 6.5 search\_students

This api action searches students / trades for provided query.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/search\_students/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it can be a name of subject.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <students\_count>, “students” : <students\_details>}<students\_details> (json array) : This is an array of json object. Each object provides details of students and has structure as follow : <students\_count> (integer) : Count of total search result. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

## 6.6 search\_category

This api action searches category for provided query.

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/search/search\_category/<auth\_token>/<query> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | query | string |

query is a search query string. More specifically, it can be a name of subject.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "search\_count" : <categories\_count>, “categories” : <categories\_details>}<category\_details> (json array) : This is an array of json object. Each object provides details of subject and has structure as follow : { "created\_at": <creation date and time>, "description": <subject description>, "id":<category id>, "is\_featured": <featured flag true/false>, "name": <name of subject>, "updated\_at": <last update date and time>, "\_type": "subject"}<category\_count> (integer) : Count of total search result. |
| 401 | { "status" : 401, "error" : “Invalid user. Login and try again.”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

7.Universities

This actions returns all university details

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/universities |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status" : 200, "university\_details" : <university\_details> }<university\_details> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of complete university data. Refer database schema to obtain columns as key for json object. |
| 401 | { "status" : 401, "error" : “University is not present”} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

8.Tasks

8.1 get\_tasks

This actions returns all tasks with details for the user for the current week, unless start date is specified

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/task/get\_tasks/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | start\_date (optional, default Current weeks' start date) | string(yyyy-mm-dd) |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "tasks": { "2015-01-27": [{ "id": 208, "resource\_id": 35, "resource\_type": "User", "name": "Test", "description": "Test", "start\_time": "2015-01-27T12:45:00.000-05:00", "created\_at": "2015-01-27T01:16:23.000-05:00", "updated\_at": "2015-01-27T01:16:23.000-05:00", "status": "Accepted", "frequency": null, "reminder": 1, "initiator\_id": null, "editable": true }] }, "date\_range": ["2015-01-25", "2015-01-31"]}<tasks> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of task date and all Tasks within that date, . Refer database schema to obtain columns as key for json object. |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

8.2 get\_task

This actions returns specific task details

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/task/get\_task/<auth\_token>/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | Id | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "task": { "id": 208, "resource\_id": 35, "resource\_type": "User", "name": "asfadsf", "description": "asfasdfsdaf", "start\_time": "2015-01-27T12:45:00.000-05:00", "created\_at": "2015-01-27T01:16:23.000-05:00", "updated\_at": "2015-01-27T01:16:23.000-05:00", "status": "Accepted", "frequency": null, "reminder": 1, "initiator\_id": null, "editable": true }}<task> (json object) : This is a json object. This object consists of the details of that specific id's task.  |
| 401 | {"status":401,"error":"Unauthorized"} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

**Initiator\_id: is the user\_id of the user who created the group task. Initator will be null for Personal tasks, this can be used to show the edit/delete link for that Task.**

8.3 add\_task

This actions adds a task

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/task/add\_task/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| Post | task | Hash |
| POST | group\_id (optional when task is for a group) | String |
| POST | user\_ids (optional when task is for selected list of friends) | String (comma separated friends id's) |

{

 "task":{

 "name": "Test",

 "description": "Description",

 "date": "2015-01-28",

 "stime": "5:00 AM",

 "reminder": 1

 }

}

Reminder options are in hours. Default is 1 hour.

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "message": "Task Added successfully.", "task": { "id": 212, "resource\_id": 35, "resource\_type": "User", "name": "Tes", "description": "Description", "start\_time": "2015-01-28T05:00:00.000-05:00", "created\_at": "2015-01-27T05:14:10.539-05:00", "updated\_at": "2015-01-27T05:14:10.539-05:00", "status": "Accepted", "frequency": null, "reminder": 1, “initiator\_id”: 65 }}<task> (json object) : This is a json object. This object consists of the details of the new task.  |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

8.4 update\_task

This actions update a task

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/task/update\_task/<auth\_token>/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| Post | task | Hash |
| POST | group\_id (optional when task is for a group) | String |
| POST | user\_ids (optional when task is for selected list of friends) | String (comma separated friends id's) |

{

 "task":{

 "name": "Test",

 "description": "Description",

 "date": "2015-01-28",

 "stime": "5:00 AM",

 "reminder": 1

 }

}

Reminder options are in hours. Default is 1 hour.

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "message": "Task updated successfully.", "task": { "id": 212, "resource\_id": 35, "resource\_type": "User", "name": "Tes", "description": "Description", "start\_time": "2015-01-28T05:00:00.000-05:00", "created\_at": "2015-01-27T05:14:10.539-05:00", "updated\_at": "2015-01-27T05:14:10.539-05:00", "status": "Accepted", "frequency": null, "reminder": 1, “initiator\_id”: 65 }}<task> (json object) : This is a json object. This object consists of the details of the new task.  |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

8.5 remove\_task

This actions update a task

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/task/update\_task/<auth\_token>/id |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {"status":200,"message":"Task removed successfully."} |
| 401 | {"status":401,"error":"Unauthorized"} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

9.Events

9.1 get\_events

This actions returns all tasks with details for the user for the current week, unless start date is specified

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/event/get\_events/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | start\_date (optional, default Current weeks' start date) | string(yyyy-mm-dd) |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "events": { "2015-01-27": [{ "id": 208, "resource\_id": 35, "resource\_type": "User", "name": "Test", "description": "Test", "start\_time": "2015-01-27T12:45:00.000-05:00", "created\_at": "2015-01-27T01:16:23.000-05:00", "updated\_at": "2015-01-27T01:16:23.000-05:00", "status": "Accepted", "frequency": null, "reminder": 1, "initiator\_id": null }] }, "date\_range": ["2015-01-25", "2015-01-31"]}<events> (json array) : This is an array of json object. Each object (database tuples are represented as json object) consists of event date and all Events within that date, . Refer database schema to obtain columns as key for json object. |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

9.2 get\_event

This actions returns specific event details

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **GET** | mobile\_app\_api/v1/event/get\_event/<auth\_token>/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | auth\_token | string |
| GET | Id | string |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "event": { "id": 208, "resource\_id": 35, "resource\_type": "User", "name": "asfadsf", "description": "asfasdfsdaf", "start\_time": "2015-01-27T12:45:00.000-05:00", "created\_at": "2015-01-27T01:16:23.000-05:00", "updated\_at": "2015-01-27T01:16:23.000-05:00", "status": "Accepted", "frequency": null, "reminder": 1, "initiator\_id": null }}<event> (json object) : This is a json object. This object consists of the details of that specific id's event.  |
| 401 | {"status":401,"error":"Unauthorized"} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

**Initiator\_id: is the user\_id of the user who created the group event. Initator will be null for Personal events, this can be used to show the edit/delete link for that Event.**

9.3 add\_event

This actions adds a event

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/event/add\_event/<auth\_token> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| Post | event | Hash |
| POST | group\_id (optional when task is for a group) | String |
| POST | user\_ids (optional when event is for selected list of friends) | String (comma separated friends id's) |

{

 "event":{

 "name": "Test",

 "description": "Description",

 "date": "2015-01-28",

 "stime": "5:00 AM",

 "reminder": 1

 }

}

Reminder options are in hours. Default is 1 hour.

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "message": "Event Added successfully.", "event": { "id": 212, "resource\_id": 35, "resource\_type": "User", "name": "Tes", "description": "Description", "start\_time": "2015-01-28T05:00:00.000-05:00", "created\_at": "2015-01-27T05:14:10.539-05:00", "updated\_at": "2015-01-27T05:14:10.539-05:00", "status": "Accepted", "frequency": null, "reminder": 1, “initiator\_id”: 65 }}<event> (json object) : This is a json object. This object consists of the details of the new event.  |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

9.4 update\_event

This actions updates a event

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/event/update\_event/<auth\_token>/id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| Post | event | Hash |
| POST | group\_id (optional when event is for a group) | String |
| POST | user\_ids (optional when event is for selected list of friends) | String (comma separated friends id's) |

{

 "event":{

 "name": "Test",

 "description": "Description",

 "date": "2015-01-28",

 "stime": "5:00 AM",

 "reminder": 1

 }

}

Reminder options are in hours. Default is 1 hour.

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "status": 200, "message": "Event updated successfully.", "event": { "id": 212, "resource\_id": 35, "resource\_type": "User", "name": "Tes", "description": "Description", "start\_time": "2015-01-28T05:00:00.000-05:00", "created\_at": "2015-01-27T05:14:10.539-05:00", "updated\_at": "2015-01-27T05:14:10.539-05:00", "status": "Accepted", "frequency": null, "reminder": 1, “initiator\_id”: 65 }}<task> (json object) : This is a json object. This object consists of the details of the updated event.  |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |

9.5 remove\_event

This actions remove a event

### Request

|  |  |
| --- | --- |
| **Method** | **URL**  |
| **POST** | mobile\_app\_api/v1/event/remove\_event/<auth\_token>/id |

**Response**

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {"status":200,"message":"Event removed successfully."} |
| 401 | {"status":401,"error":"Unauthorized"} |
| 500 | { "status" : 500, "error" : "Something went wrong. Please try again later."} |