```
Title
                         Show all Addresses
URL
                         /api/addresses
Method
                         GET
URL Params
                         N/A
Headers
                         { auth-token : <<Your with token>> }
Success Response
                           "id": 6,
                           "longitude": -9.1371262,
                           "latitude": 38.7271338,
                           "street": null
                          },
                          {
                           "id": 7,
                           "longitude": -8.3673551,
                           "latitude": 41.4207052,
                           "street": null
                          },
                          {
                           "id": 8,
                           "longitude": -8.3673551,
                           "latitude": 41.4207052,
                           "street": null
                          },
                           "id": 9,
                           "longitude": -9.2420951,
                           "latitude": 38.7287289,
                           "street": null
                          },
                          {
                           "id": 10,
                           "longitude": -9.1484923,
                           "latitude": 38.7234688,
                           "street": null
                          },
                           "id": 11,
                           "longitude": -9.24167,
                           "latitude": 38.72667,
                           "street": "Rua Rua João Paulo II, Azambuja, Lisbon, Portugal"
                          },
                           "id": 12,
                           "longitude": -9.1369241,
                           "latitude": 38.7270066,
                           "street": "Rua Jacinta Marto, Anjos, Arroios, Portugal"
                          },
                          {
                           "id": 13,
                           "longitude": -1.219482,
```

	"latitude": 47.41322,  "street": "" } ]
Error Response	401 - "Invalid Token"
Sample call	<pre>await axios.get('/api/addresses')</pre>

Title	Get Address By Id
	,
URL	/api/addresses/:id
Method	GET
URL Params	Required:
	id=[integer]
	example: id=12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Success Response	{
	"id": 6,
	"longitude": -9.1371262,
	"latitude": 38.7271338,
	"street": null
	}
Error Response	401 - "Invalid Token"
Sample call	<pre>await axios.get('/api/addresses/1')</pre>

Title	Create Address
URL	/api/addresses/
Method	POST
URL	N/A
Params	
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{   "longitude": -9.1369241,   "latitude": 38.7270066,   "street": "Rua Jacinta Marto, Anjos, Arroios, Portugal" },
Success Respons e	{     "id": 12,     "longitude": -9.1369241,     "latitude": 38.7270066,     "street": "Rua Jacinta Marto, Anjos, Arroios, Portugal" },
Error Respons e	401 - "Invalid Token" 400 – "Bad Request – Data is invalid"
Sample call	<pre>await axios.post('/api/addresses', {     "longitude": -9.1369241,     "latitude": 38.7270066,</pre>

```
"street": "Rua Jacinta Marto, Anjos, Arroios, Port
ugal"
     },
)
```

Title	Update Address
URL	/api/addresses/:id
Method	PUT
URL Params	Required:
	id=[integer]
	example: id=12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"longitude": -9.1369241,
	"latitude": 38.7270066,
	"street": "Rua Jacinta Marto, Anjos, Arroios, Portugal"
	},
Success Response	{
	"id": 12,
	"longitude": -9.1369241,
	"latitude": 38.7270066,
	"street": "Rua Jacinta Marto, Anjos, Arroios, Portugal"
	},
Error Response	401 - "Invalid Token"
Sample call	<pre>await axios.put('/api/addresses/1')</pre>

Title	Delete Address
URL	/api/addresses/:id
Method	DELETE
URL Params	Required:
	id=[integer]
	example: id=12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{
	"id": 12,
	"longitude": -9.1369241,
	"latitude": 38.7270066,
	"street": "Rua Jacinta Marto, Anjos, Arroios, Portugal"
	},
Error Response	401 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/addresses/1')</pre>

Title	Login
URL	/api/authentication/login

Method	POST
URL Params	N/A
Headers	N/A
Data	{   "email":"vasco@gmail.com",   "password":"123testeteste123" }
Success Response	{     "token": < <jwt token="">&gt; },</jwt>
Error Response	400 - "Email or password is wrong"
Sample call	<pre>await axios.post('/api/authentication/login')</pre>

Title	Register
URL	/api/authentication/register
Method	POST
URL Params	N/A
Headers	N/A
Data	{   "email":"vasco@gmail.com",   "password":"123testeteste123",   "firstName":"Vasco",   "lastName":"Pereira" }
Success Response	{     "user": "12" // NEW USER ID     },
Error Response	
Sample call	<pre>await axios.post('/api/authentication/register')</pre>

Title	Encrypt
URL	/api/authentication/encrypt/:password
Method	GET
URL Params	Required:
	password =[String]
	example: password ="123teste"
Headers	N/A
Data	{
	"email":"vasco@gmail.com",
	"password":"123testeteste123",
	"firstName":"Vasco",
	"lastName":"Pereira"
	}
Success Response	{
	"password": < <encrypted password="">&gt;</encrypted>
	},
Error Response	500 – "Failed to encrypt"

Sample call	<pre>await axios.get('/api/authentication/encrypt/123teste')</pre>

Title	Get All Modules
URL	/api/modules/
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	<pre>[</pre>
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/modules/')</pre>

Title	Get Module By Id
URL	/api/modules/:id
Method	GET
URL Params	Required:
	id=[integer]
	example: id=12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{
	"id": 1,
	"title": "MVP",
	"collapsed": 1,
	"project": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/modules/1')</pre>

Title	Get Modules By Project
URL	/api/modules/getByProject/:projectId
Method	GET

URL Params	Required: projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	[{     "id": 1,     "title": "MVP",     "collapsed": 1,     "project": 1 }]
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/modules/getByProject/1')</pre>

Title	Add Module
URL	/api/modules/
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{   "title": "MVP",   "collapsed": 1,   "project": 1 }
Success Response	{     "id": 1,     "title": "MVP",     "collapsed": 1,     "project": 1 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/modules/')</pre>

Title	Update Module
URL	/api/modules/1
Method	PUT
URL Params	Required:
	projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"id": 1,
	"title": "MVP",
	"collapsed": 1,
	"project": 1
	}
Success Response	{

Title	Doloto Modulo
Title	Delete Module
URL	/api/modules/1
Method	DELETE
URL Params	Required:
	projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{
	"id": 1,
	"title": "MVP",
	"collapsed": 1,
	"project": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/modules/1')</pre>

Title	Get All Projects
URL	/api/projects
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/a
Success Response	<pre>[     "id": 1,     "name": "Lava Lamp",     "description": "3Y1S Project IADE",     "owner": 1     },     {         "id": 2,         "name": "Projeto Teste",         "description": null,         "owner": 3     },     {         "id": 22,         "name": "DEMO",     } }</pre>

	"description": null,
	"owner": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/projects')</pre>

Title	Get By Id
URL	/api/projects/:id
Method	GET
URL Params	Required:
	projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{   "id": 1,   "name": "Lava Lamp",   "description": "3Y1S Project IADE",   "owner": 1 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/projects/1')</pre>

Title	Add Project
URL	/api/projects
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{     "name": "Lava Lamp",     "description": "3Y1S Project IADE",     "owner": 1 }
Success Response	{   "id": 1,   "name": "Lava Lamp",   "description": "3Y1S Project IADE",   "owner": 1 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/projects/');</pre>

Title	Update Project
URL	/api/projects/:id
Method	PUT
URL Params	Required:
	projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{   "id": 1,   "name": "Lava Lamp",   "description": "3Y1S Project IADE",   "owner": 1 }
Success Response	{   "id": 1,   "name": "Lava Lamp",   "description": "3Y1S Project IADE",   "owner": 1 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/projects/1')</pre>

Title	Delete Project
URL	/api/projects/:id
Method	DELETE
URL Params	Required:
	projectId =[integer]
	example: projectId =12
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"id": 1,
	"name": "Lava Lamp",
	"description": "3Y1S Project IADE",
	"owner": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/projects/1')</pre>

```
Title
                         Get all Project Users
URL
                         /api/projectsUsers/
Method
                         GET
URL Params
                         N/A
                         { auth-token : <<Your with token>> }
Headers
Data
                         N/A
Success Response
                         [
                           "project": 1,
                           "user": 1,
                           "role": "Manager",
                           "isCurrent": {
                            "type": "Buffer",
                            "data": [
                             1
                           }
                          },
                           "project": 1,
                           "user": 3,
                           "role": "Programmer",
                           "isCurrent": {
                            "type": "Buffer",
                            "data": [
                             1
                           }
                          },
                           "project": 1,
                           "user": 10,
                           "role": "Tester",
                           "isCurrent": {
                            "type": "Buffer",
                            "data": [
                             1
                            ]
                           }
                          },
                           "project": 2,
                           "user": 1,
                           "role": "Manager",
                           "isCurrent": {
                            "type": "Buffer",
                            "data": [
                             0
```

Title	Get Project For Current User
URL	/api/projectsUsers/getCurrentForUser
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{     "project": 1,     "user": 1,     "role": "Manager",     "isCurrent": {         "type": "Buffer",         "data": [
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/projectUsers/getCurrentForUser')</pre>

Title	Get all Projects of Current User
URL	/api/projectsUsers/getByCurrentUser
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"project": {
	"id": 1,

```
"name": "Lava Lamp",
   "description": "3Y1S Project IADE",
   "owner": 1
  },
  "user": {
   "id": 1,
   "email": "vascop_98@hotmail.com",
   "firstName": "Vasco",
   "lastName": "Pereira",
   "password":
"$2a$10$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
  "role": "Manager",
  "isCurrent": {
   "type": "Buffer",
   "data": [
    1
  ]
 }
},
  "project": {
   "id": 2,
   "name": "Projeto Teste",
   "description": null,
   "owner": 3
  },
  "user": {
   "id": 1,
   "email": "vascop_98@hotmail.com",
   "firstName": "Vasco",
   "lastName": "Pereira",
   "password":
"$2a$10$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
  "role": "Manager",
  "isCurrent": {
   "type": "Buffer",
   "data": [
    0
  ]
 }
},
  "project": {
   "id": 22,
   "name": "DEMO",
   "description": null,
   "owner": 1
  },
  "user": {
```

```
"id": 1,
                     "email": "vascop_98@hotmail.com",
                    "firstName": "Vasco",
                     "lastName": "Pereira",
                     "password":
                  "$2a$10$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
                   },
                    "role": null,
                    "isCurrent": {
                    "type": "Buffer",
                    "data": [
                     0
                  }
Error Response
                  400 - "Invalid Token"
Sample call
                 await axios.get('/api/projectUsers/getByCurrentUser)
```

Title	Get By Project
URL	/api/projectsUsers/getByProject/:id/:deep?
Method	GET
URL Params	Required:
	projectId =[integer]
	example: projectId =12
	Not Required:
	Deep=[boolean]
	Exemple deep=true
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success	]
Response	{
	"project": {
	"id": 1,
	"name": "Lava Lamp",
	"description": "3Y1S Project IADE",
	"owner": 1
	},
	"user": {
	"id": 1,
	"email": "vascop_98@hotmail.com",
	"firstName": "Vasco",
	"lastName": "Pereira",
	"password":
	"\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
	},
	"role": "Manager",
	"isCurrent": {

```
"type": "Buffer",
   "data": [
    1
  ]
 }
},
  "project": {
   "id": 1,
   "name": "Lava Lamp",
   "description": "3Y1S Project IADE",
   "owner": 1
  "user": {
  "id": 3,
   "email": "ruca",
   "firstName": "João",
   "lastName": "Calapez",
   "password":
"$2a$10$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO"
  "role": "Programmer",
  "isCurrent": {
   "type": "Buffer",
  "data": [
  ]
 }
},
  "project": {
  "id": 1,
   "name": "Lava Lamp",
   "description": "3Y1S Project IADE",
   "owner": 1
  },
  "user": {
  "id": 10,
   "email": "teste",
   "firstName": "teste",
   "lastName": "teste",
   "password":
"$2a$10$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO"
 },
  "role": "Tester",
  "isCurrent": {
   "type": "Buffer",
   "data": [
    1
  ]
```

	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/projectUsers/getByProject/1/deep')</pre>

```
Title
                   Get By User
                   /api/projectsUsers/getByUser/: userId /:deep?
URL
Method
                   GET
URL Params
                   Required:
                   userId =[integer]
                   example: projectId =12
                   Not Required:
                   Deep=[boolean]
                   Exemple deep=true
                   { auth-token : << Your with token>> }
Headers
Data
                   N/A
Success Response
                     "project": {
                      "id": 1,
                      "name": "Lava Lamp",
                      "description": "3Y1S Project IADE",
                      "owner": 1
                     },
                     "user": {
                      "id": 1,
                      "email": "vascop_98@hotmail.com",
                      "firstName": "Vasco",
                      "lastName": "Pereira",
                      "password":
                   "$2a$10$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
                     },
                     "role": "Manager",
                     "isCurrent": {
                      "type": "Buffer",
                      "data": [
                       1
                      ]
                     }
                    },
                     "project": {
                      "id": 1,
                      "name": "Lava Lamp",
                      "description": "3Y1S Project IADE",
                      "owner": 1
                     },
                     "user": {
                      "id": 3,
```

```
"email": "ruca",
                      "firstName": "João",
                      "lastName": "Calapez",
                      "password":
                   "$2a$10$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO"
                     "role": "Programmer",
                     "isCurrent": {
                      "type": "Buffer",
                      "data": [
                       1
                     ]
                    }
                   },
                     "project": {
                      "id": 1,
                      "name": "Lava Lamp",
                      "description": "3Y1S Project IADE",
                      "owner": 1
                     },
                     "user": {
                      "id": 10,
                      "email": "teste",
                      "firstName": "teste",
                      "lastName": "teste",
                      "password":
                   "$2a$10$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO"
                     },
                     "role": "Tester",
                     "isCurrent": {
                      "type": "Buffer",
                      "data": [
                       1
                      ]
                    }
                   }
                   400 - "Invalid Token"
Error Response
Sample call
                  await axios.get('/api/projectUsers/getByUser/1/deep')
```

Title	Add Project User
URL	/api/projectUsers
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"project": 1,
	"user": 1,

```
"role": "Manager",
                         "isCurrent": 1
                         }
Success Response
                         "project": 1,
                         "user": 1,
                         "role": "Manager",
                         "isCurrent": {
                          "type": "Buffer",
                          "data": [
                          1
                          ]
                       400 - "Invalid Token"
Error Response
Sample call
                       await axios.post('/api/projectUsers/')
```

Title	Update Project User
URL	/api/projectUsers/
Method	PUT
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{     "project": 1,     "user": 1,     "role": "Manager",     "isCurrent": 1   }
Success Response	{     "project": 1,     "user": 1,     "role": "Manager",     "isCurrent": {       "type": "Buffer",       "data": [           1       ]     }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/projectUsers/')</pre>

Title	Delete Project User
URL	/api/projectUsers/:projectId/:userId
Method	DELETE

URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	Required:
	projectId [Integer]
	userId [Integer]
Success Response	
	{
	"project": 1,
	"user": 1,
	"role": "Manager",
	"isCurrent": {
	"type": "Buffer",
	"data": [
	1
	1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/projectUsers/1/2')</pre>

Title	Delete Project User
URL	/api/projectUsers/:projectId/:userId
Method	DELETE
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	Required: projectId [Integer] userId [Integer]
Success Response	{     "project": 1,     "user": 1,     "role": "Manager",     "isCurrent": {         "type": "Buffer",         "data": [         1      ]     }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/projectUsers/1/2')</pre>

Title	Get All Tasks
URL	/api/tasks/
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	

```
Success Response
                           "id": 1,
                           "module": 1,
                           "title": "Modules",
                           "dueDate": "2020-12-05T00:00:00.000Z",
                           "owner": 3.
                           "approver": 1,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": 11
                          },
                           "id": 5,
                           "module": 1,
                           "title": "Task Assignment",
                           "dueDate": "2020-12-06T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                          {
                           "id": 6,
                           "module": 1,
                           "title": "Users",
                           "dueDate": "2020-12-07T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "High",
                           "state": "Complete",
                           "address": null
                          },
                          {
                           "id": 7,
                           "module": 1,
                           "title": "Teams",
                           "dueDate": "2020-12-09T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                           "id": 9,
                           "module": 1,
                           "title": "Tasks",
                           "dueDate": "2020-12-02T00:00:00.000Z",
                           "owner": 1,
```

```
"approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
{
 "id": 21,
 "module": 1,
 "title": "Gantt Chart",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Complete",
 "address": 12
},
 "id": 22,
 "module": 1,
 "title": "Task Flow",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
 "id": 23,
 "module": 1,
 "title": "Standardize Server Side",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": 1,
 "priority": "Low",
 "state": "To Do",
 "address": null
},
 "id": 24,
 "module": 1,
 "title": "GitHub API",
 "dueDate": "2020-12-14T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Doing",
 "address": null
},
{
 "id": 25,
```

```
"module": 1,
                           "title": "Leaflet API",
                           "dueDate": "2020-12-14T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                           "id": 31,
                           "module": 8,
                           "title": "Testar Gantt",
                           "dueDate": "2021-01-13T00:00:00.000Z",
                           "owner": null,
                          "approver": null,
                           "priority": null,
                           "state": "Unassigned",
                           "address": null
                          },
                          {
                           "id": 32,
                           "module": 1,
                           "title": "Multiple Projects",
                           "dueDate": "2021-01-06T00:00:00.000Z",
                           "owner": 1,
                           "approver": 1,
                           "priority": "High",
                           "state": "Complete",
                           "address": null
                         }
Error Response
                         400 - "Invalid Token"
Sample call
                        await axios.get('/api/tasks')
```

Title	Get Task by Id
URL	/api/tasks/1
Method	GET
URL Params	Required:
	Id: [Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	
Success Response	{     "id": 1,     "module": 1,     "title": "Modules",     "dueDate": "2020-12-05T00:00:00.000Z",     "owner": 3,     "approver": 1,

	"priority": "Medium", "state": "Complete", "address": 11 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/tasks/1')</pre>

Title	Add Task
URL	/api/tasks/
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{     "module": 1,     "title": "Modules",     "dueDate": "2020-12-05T00:00:00.000Z",     "owner": 3,     "approver": 1,     "priority": "Medium",     "state": "Complete",     "address": 11 }
Success Response	{     "id": 1,     "module": 1,     "title": "Modules",     "dueDate": "2020-12-05T00:00:00.000Z",     "owner": 3,     "approver": 1,     "priority": "Medium",     "state": "Complete",     "address": 11 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/tasks/')</pre>

Title	Update Task
URL	/api/tasks/:id
Method	PUT
URL Params	Required:
	Id=[integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{     "id": 1,     "module": 1,     "title": "Modules",     "dueDate": "2020-12-05T00:00:00.000Z",

```
"owner": 3,
                         "approver": 1,
                         "priority": "Medium",
                         "state": "Complete",
                         "address": 11
Success Response
                         "id": 1,
                         "module": 1,
                         "title": "Modules",
                         "dueDate": "2020-12-05T00:00:00.000Z",
                         "owner": 3,
                         "approver": 1,
                         "priority": "Medium",
                         "state": "Complete",
                         "address": 11
Error Response
                        400 - "Invalid Token"
Sample call
                       await axios.put('/api/tasks/1')
```

Title	Delete Task
URL	/api/tasks/:id
Method	DELETE
URL Params	Required:
	Id=[integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{     "id": 1,     "module": 1,     "title": "Modules",     "dueDate": "2020-12-05T00:00:00.000Z",     "owner": 3,     "approver": 1,     "priority": "Medium",     "state": "Complete",     "address": 11 }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/tasks/1')</pre>

Title	Get Tasks By Module
URL	/api/tasks/getByModule/:moduleId
Method	GET
URL Params	Required:
	ModuleId=[integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>

```
Data
                         N/A
Success Response
                         [
                          {
                           "id": 1,
                           "module": 1,
                           "title": "Modules",
                           "dueDate": "2020-12-05T00:00:00.000Z",
                           "owner": 3,
                           "approver": 1,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": 11
                          },
                          {
                           "id": 5,
                           "module": 1,
                           "title": "Task Assignment",
                           "dueDate": "2020-12-06T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                          {
                           "id": 6,
                           "module": 1,
                           "title": "Users",
                           "dueDate": "2020-12-07T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "High",
                           "state": "Complete",
                           "address": null
                          },
                           "id": 7,
                           "module": 1,
                           "title": "Teams",
                           "dueDate": "2020-12-09T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                           "id": 9,
                           "module": 1,
                           "title": "Tasks",
                           "dueDate": "2020-12-02T00:00:00.000Z",
```

```
"owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
{
 "id": 21,
 "module": 1,
 "title": "Gantt Chart",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Complete",
 "address": 12
},
 "id": 22,
 "module": 1,
 "title": "Task Flow",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
{
 "id": 23,
 "module": 1,
 "title": "Standardize Server Side",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
"approver": 1,
 "priority": "Low",
 "state": "To Do",
 "address": null
},
{
 "id": 24,
 "module": 1,
 "title": "GitHub API",
 "dueDate": "2020-12-14T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Doing",
 "address": null
},
```

```
"id": 25,
                           "module": 1,
                           "title": "Leaflet API",
                           "dueDate": "2020-12-14T00:00:00.000Z",
                           "owner": 1,
                           "approver": null,
                           "priority": "Medium",
                           "state": "Complete",
                           "address": null
                          },
                           "id": 32,
                           "module": 1,
                           "title": "Multiple Projects",
                           "dueDate": "2021-01-06T00:00:00.000Z",
                           "owner": 1,
                           "approver": 1,
                           "priority": "High",
"state": "Complete",
                           "address": null
                          }
Error Response
                         400 - "Invalid Token"
                        await axios.get('/api/tasks/getByModule/1')
Sample call
```

Title	Get Tasks By Project
URL	/api/tasks/getByModule/:projectId
Method	GET
URL Params	Required:
	projectId =[integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	[
	{
	"id": 1,
	"module": 1,
	"title": "Modules",
	"dueDate": "2020-12-05T00:00:00.000Z",
	"owner": 3,
	"approver": 1,
	"priority": "Medium",
	"state": "Complete",
	"address": 11
	},
	{
	"id": 5,
	"module": 1,
	"title": "Task Assignment",
	"dueDate": "2020-12-06T00:00:00.000Z",

```
"owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Complete",
 "address": null
},
{
 "id": 6,
 "module": 1,
 "title": "Users",
 "dueDate": "2020-12-07T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
 "id": 7,
 "module": 1,
 "title": "Teams",
 "dueDate": "2020-12-09T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
"state": "Complete",
 "address": null
},
{
 "id": 9,
 "module": 1,
 "title": "Tasks",
 "dueDate": "2020-12-02T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
{
 "id": 21,
 "module": 1,
 "title": "Gantt Chart",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Complete",
 "address": 12
},
```

```
"id": 22,
 "module": 1,
 "title": "Task Flow",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "High",
 "state": "Complete",
 "address": null
},
 "id": 23,
 "module": 1,
 "title": "Standardize Server Side",
 "dueDate": "2020-12-12T00:00:00.000Z",
 "owner": 1,
 "approver": 1,
 "priority": "Low",
 "state": "To Do",
 "address": null
},
 "id": 24,
 "module": 1,
 "title": "GitHub API",
 "dueDate": "2020-12-14T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Doing",
 "address": null
},
 "id": 25,
 "module": 1,
 "title": "Leaflet API",
 "dueDate": "2020-12-14T00:00:00.000Z",
 "owner": 1,
 "approver": null,
 "priority": "Medium",
 "state": "Complete",
 "address": null
},
{
 "id": 32,
 "module": 1,
 "title": "Multiple Projects",
 "dueDate": "2021-01-06T00:00:00.000Z",
 "owner": 1,
 "approver": 1,
 "priority": "High",
```

```
Get All Teams
Title
URL
                         /api/teams
Method
                         GET
URL Params
                         N/A
                         { auth-token : << Your with token>> }
Headers
Data
                         N/A
Success Response
                         [
                           "id": 1,
                           "name": "Dev Team",
                           "project": 1,
                           "description": null,
                           "leader": 1
                          },
                           "id": 2,
                           "name": "UX Team",
                           "project": 1,
                           "description": null,
                           "leader": 3
                          },
                          {
                           "id": 3,
                           "name": "QA Team",
                           "project": 1,
                           "description": null,
                           "leader": 1
                          },
                          {
                           "id": 4,
                           "name": "Management",
                           "project": 1,
                           "description": null,
                           "leader": 1
                          },
                           "id": 11,
                           "name": "Test",
                           "project": 2,
                           "description": null,
                           "leader": null
```

```
},
{
    "id": 12,
    "name": "Teste",
    "project": 2,
    "description": null,
    "leader": null
    }
]

Error Response     400 - "Invalid Token"

Sample call

await axios.get(\'/api/teams')
```

Title	Get Team By Id
URL	/api/teams/:id
Method	GET
URL Params	Required:
	ld:Integer
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{
	"id": 1,
	"name": "Dev Team",
	"project": 1,
	"description": null,
	"leader": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/teams/1')</pre>

Title	Get Teams By Project
URL	/api/teams/getByProject/:id
Method	GET
URL Params	Required:
	ld:Integer
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	[

```
"project": 1,
                          "description": null,
                          "leader": 3
                         },
                          "id": 3,
                         "name": "QA Team",
                          "project": 1,
                          "description": null,
                          "leader": 1
                         },
                          "id": 4,
                          "name": "Management",
                         "project": 1,
                          "description": null,
                          "leader": 1
                        }
Error Response
                        400 - "Invalid Token"
Sample call
                       await axios.get('/api/teams/getByProject/1')
```

Title	Update Team
URL	/api/teams/:id
Method	PUT
URL Params	Required:
	ld:Integer
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"id": 1,
	"name": "Dev Team",
	"project": 1,
	"description": null,
	"leader": 1
	}
Success Response	{
	"id": 1,
	"name": "Dev Team",
	"project": 1,
	"description": null,
	"leader": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/teams/1')</pre>

Title	Create Team
URL	/api/teams/

Method	PUT
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"name": "Dev Team",
	"project": 1,
	"description": null,
	"leader": 1
	}
Success Response	{
	"id": 1,
	"name": "Dev Team",
	"project": 1,
	"description": null,
	"leader": 1
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/teams/')</pre>

Title	Get All Team Users
URL	/api/teamsUsers/
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	<pre>[ {     "user": 1,     "team": 1,     "role": "Developer" }, {     "user": 3,     "team": 1,     "role": "Developer" }, {     "user": 3,     "team": 2,     "role": null } </pre>
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/teamUsers')</pre>

Title	Get Team Users By Team
URL	/api/teamsUsers/getByTeam/:teamid

Method	GET
URL Params	Required:
	Teamid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	},
	{
	"user": 3,
	"team": 1,
	"role": "Developer"
	},
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/teamUsers/getByTeam/1')</pre>

Title	Get Team Users By User
URL	/api/teamsUsers/getByUser/:userid
Method	GET
URL Params	Required:
	userid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
	]
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/teamUsers/getByUser/1')</pre>

Title	Get Team Users By User And Team
URL	/api/teamsUsers/:teamid /:userid
Method	GET
URL Params	Required:
	userid:[Integer]
	teamid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{
	"user": 1,

	"team": 1, "role": "Developer" }
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/teamUsers/1/1')</pre>

Title	Create Team User
URL	/api/teamsUsers/
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
Success Response	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/teamUsers/')</pre>

Title	Update Team User
URL	/api/teamsUsers/:teamid /:userid
Method	PUT
URL Params	Required:
	userid:[Integer]
	teamid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
Success Response	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/teamUsers/')</pre>

Title	Delete Team Users By User
URL	/api/teamsUsers/:userid
Method	DELETE
URL Params	Required:
	userid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	
Success Response	{
	"affectedRows":2
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/teamUsers/1/')</pre>

Title	Delete Team Users By Team
URL	/api/teamsUsers/:teamid
Method	DELETE
URL Params	Required:
	teamid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	
Success Response	{
	"affectedRows":2
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/teamUsers/1')</pre>

Title	Delete Team User
URL	/api/teamsUsers/:teamid/:userId
Method	DELETE
URL Params	Required:
	teamid:[Integer]
	userid:[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	
Success Response	{
	"user": 1,
	"team": 1,
	"role": "Developer"
	}
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.delete('/api/teamUsers/1/1')</pre>

Title	Get All Users
URL	/api/users
Method	GET

URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	[  "id": 1,  "email": "vascop_98@hotmail.com",  "firstName": "Vasco",  "lastName": "Pereira",  "password":  "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm" },  {  "id": 3,  "email": "ruca",  "firstName": "João",  "lastName": "Calapez",  "password":  "\$2a\$10\$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO" },  {  "id": 10,  "email": "teste",  "firstName": "teste",  "lastName": "teste",  "lastName": "teste",  "password":  "\$2a\$10\$hqmWSSOvgV8XJJLo2gcW8ualr686GuP09qBO8txKDbvH6LlkcTpPO" } ]
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/users/')</pre>

Title	Get Current User
URL	/api/users/self
Method	GET
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	{     "id": 1,     "email": "vascop_98@hotmail.com",     "firstName": "Vasco",     "lastName": "Pereira",     "password":     "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"     },
Error Response	400 - "Invalid Token"

Sample call	<pre>await axios.get('/api/users/self')</pre>

F	
Title	Get User By Id
URL	/api/users/:id
Method	GET
URL Params	Required:
	Id=[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"id": 1,
	"email": "vascop_98@hotmail.com",
	"firstName": "Vasco",
	"lastName": "Pereira",
	"password":
	"\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
	},
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.get('/api/users/1)</pre>

Title	Create User
URL	/api/users/
Method	POST
URL Params	N/A
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{     "email": "vascop_98@hotmail.com",     "firstName": "Vasco",     "lastName": "Pereira",     "password":     "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"     },
Success Response	{     "id": 1,     "email": "vascop_98@hotmail.com",     "firstName": "Vasco",     "lastName": "Pereira",     "password":     "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"     },
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.post('/api/users/)</pre>

Title	Update User
URL	/api/users/:id
Method	PUT
URL Params	Required:
	Id=[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	{   "id": 1,   "email": "vascop_98@hotmail.com",   "firstName": "Vasco",   "lastName": "Pereira",   "password":   "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"   },
Success Response	{   "id": 1,   "email": "vascop_98@hotmail.com",   "firstName": "Vasco",   "lastName": "Pereira",   "password":   "\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"   },
Error Response	400 - "Invalid Token"
Sample call	<pre>await axios.put('/api/users/1)</pre>

Title	Delete User
URL	/api/users/:id
Method	DELETE
URL Params	Required:
	Id=[Integer]
Headers	{ auth-token : < <your token="" with="">&gt; }</your>
Data	N/A
Success Response	
	{
	"id": 1,
	"email": "vascop_98@hotmail.com",
	"firstName": "Vasco",
	"lastName": "Pereira",
	"password":
	"\$2a\$10\$TN9HDFVqRgro00ibtT.tqucM2wVR9J5wyYH4kbdx9QH0tCuOiwkNm"
	},
Error Response	400 - "Invalid Token"

Sample call	<pre>await axios.delete('/api/users/1)</pre>