

# Teammee API

Move your app forward with the Teammee API

**Version** 1.0.2

## Contact information

info@teammee.com (mailto:info@teammee.com)

<http://www.teammee.com/> (http://www.teammee.com/)

## Security

teammee\_application (OAuth 2.0)

Authenticate

### OAuth2 Client API access

Flow application

Token <http://nomeinstanza.teamee.com/oauth/token>  
URL (<http://nomeinstanza.teamee.com/oauth/token>)

### Scopes

|                           |                            |
|---------------------------|----------------------------|
| <b>read_post</b>          | Grant read posts           |
| <b>read_event</b>         | Grant read events          |
| <b>read_document</b>      | Grant read documents       |
| <b>read_task</b>          | Grant read tasks           |
| <b>read_livestream</b>    | Grant read Livestreams     |
| <b>add_post</b>           | Grant write post           |
| <b>add_event</b>          | Grant write event          |
| <b>add_document</b>       | Grant write document       |
| <b>add_task</b>           | Grant write task           |
| <b>delete_post</b>        | Grant delete post item     |
| <b>delete_event</b>       | Grant delete event item    |
| <b>delete_document</b>    | Grant delete document item |
| <b>delete_task</b>        | Grant delete task item     |
| <b>modify_post</b>        | Grant modify post item     |
| <b>modify_event</b>       | Grant modify event item    |
| <b>modify_document</b>    | Grant modify document item |
| <b>modify_task</b>        | Grant modify task item     |
| <b>impersonate_any</b>    | Grant impersonate any user |
| <b>add_friend</b>         | Grant add connection       |
| <b>delete_friend</b>      | Grant remove connection    |
| <b>add_group</b>          | Grant add group            |
| <b>modify_group</b>       | Grant group modify         |
| <b>delete_group</b>       | Grant group delete         |
| <b>manage_group_users</b> | Grant manage group members |

# Paths

PUT /Post/

Posts

## Summary

Insert a Post

## Description

Insert a post with attachments, the response return ok and id of inserted post. The request must be a multipart/form-data

## Parameters

| Name         | Located in | Description   | Required | Schema    |
|--------------|------------|---|----------|-----------|
| payload.json | formData   | Payload JSON with all data for creation of a post. Must be a Json encoded string based on ItemNew model | Yes      | ⇔ string  |
| impersonate  | formData   | id of user impersonate by application (require scope impersonate_any)                                   | No       | ⇔ integer |
| fileattach   | formData   | File attached at post   | No       | ⇔ file    |

## Responses

| Code    | Description      | Schema  |
|---------|------------------|---|
| 200     | Inserted post    | ⇔<br><pre>▼ Insert {<br/>  result: ▶ string<br/>  id:    ▶ integer<br/>}</pre>                    |
| default | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error:           ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

## Security

| Security Schema    | Scopes          |
|--------------------|-----------------|
| teamme_application | <b>add_post</b> |

## Summary

Delete a post

## Description

Delete a post identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema                            |
|-------------|------------|---|----------|-----------------------------------|
| id_post     | query      | id of post to delete  | Yes      | $\Leftrightarrow$ integer (int32) |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer         |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Post deleted     | $\Leftrightarrow$ <pre>▼ Defaultmessage {   result: ▶ string }</pre>                     |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <pre>▼ Error {   error: ▶ string   error_description: ▶ string }</pre> |

## Security

| Security Schema    | Scopes             |
|--------------------|--------------------|
| teamme_application | <b>delete_post</b> |

## Summary

Modify a post

## Description

Modify a post identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema   |
|-------------|------------|---|----------|--|
| body        | body       | Payload with post model to modify                                     | Yes      | <pre>           ▼ <b>PostModify</b> {             id_post: ▶ integer             id_group: ▶ integer             isPublic: ▶ boolean             toFriends: ▶ boolean             message: ▶ string             user_list: ▶ []           }         </pre> |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | <pre>           ⇔ integer         </pre>   |

## Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Post modified    | <pre>           ▼ <b>Defaultmessage</b> {             result: ▶ string           }         </pre>                               |
| <b>default</b> | Unexpected error | <pre>           ▼ <b>Error</b> {             error: ▶ string             error_description: ▶ string           }         </pre> |

## Security

| Security Schema    | Scopes             |
|--------------------|--------------------|
| teamme_application | <b>modify_post</b> |

## Summary

Get one or more posts

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | query      | get posts in this group   | No       | ⇔ integer |
| id_post     | query      | get get post by id  | No       | ⇔ integer |
| date_from   | query      | get posts before this unixtimestamp (if 0 get lasts)                  | No       | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Post/s requested | ⇔ <pre>           ▼ [             array of posts             ► PostModel { }           ]           </pre>                    |
| <b>default</b> | Unexpected error | ⇔ <pre>           ▼ Error {             error: ► string             error_description: ► string           }           </pre> |

## Security

| Security Schema    | Scopes           |
|--------------------|------------------|
| teamme_application | <b>read_post</b> |

## Summary

Insert an Event

## Description

Insert an post with attachments, the response return ok and id of inserted event. The request must be a multipart/form-data

## Parameters

| Name         | Located in | Description  | Required | Schema    |
|--------------|------------|--|----------|-----------|
| payload.json | formData   | Payload JSON with all data for creation of an event. Must be a Json encoded string based on ItemNewEvent model | Yes      | ⇔ string  |
| impersonate  | formData   | id of user impersonate by application (require scope impersonate_any)  | No       | ⇔ integer |
| fileattach   | formData   | File attached at event   | No       | ⇔ file    |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Inserted event   | ⇔ <pre>▼ Insert {   result: ▶ string   id:    ▶ integer }</pre>                    |
| <b>default</b> | Unexpected error | ⇔ <pre>▼ Error {   error:           ▶ string   error_description: ▶ string }</pre> |

## Security

| Security Schema    | Scopes           |
|--------------------|------------------|
| teamme_application | <b>add_event</b> |

## Summary

Delete an event

## Description

Delete an event identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema                            |
|-------------|------------|---|----------|-----------------------------------|
| id_event    | query      | id of event to delete   | Yes      | $\Leftrightarrow$ integer (int32) |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer         |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Event deleted    | $\Leftrightarrow$ <pre>▼ Defaultmessage {   result: ► string }</pre>                     |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

## Security

| Security Schema    | Scopes              |
|--------------------|---------------------|
| teamme_application | <b>delete_event</b> |

## Summary

Modify an event

## Description

Modify an event identified by its id

## Parameters

| Name              | Located in | Description   | Required | Schema   |
|-------------------|------------|---|----------|--|
| body              | body       | Payload with event model to modify                                    | Yes      | <pre> ▼ EventModify {   id_event: ▶ integer   id_group: ▶ integer   isPublic: ▶ boolean   toFriends: ▶ boolean   message: ▶ string   event_from: ▶ integer   event_to: ▶ integer   webEx: ▶ boolean   user_list: ▶ [] } </pre> |
| impersonate query |            | id of user impersonate by application (require scope impersonate_any) | No       | <pre> ⇒ integer </pre>   |

## Responses

| Code    | Description      | Schema   |
|---------|------------------|--|
| 200     | Event modified   | <pre> ⇒ ▼ Defaultmessage {   result: ▶ string } </pre>                     |
| default | Unexpected error | <pre> ⇒ ▼ Error {   error: ▶ string   error_description: ▶ string } </pre> |

## Security

Security Schema

teamme\_application

Scopes

**modify\_event**

## GET /Event/

Events

### Summary

Get one or more events

### Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| nearest     | query      | get nearest events  | No       | ⇔ boolean |
| id_group    | query      | get events in this group  | No       | ⇔ integer |
| id_post     | query      | get event by id   | No       | ⇔ integer |
| date_from   | query      | get events before this unixtimestamp (if 0 get lasts)                 | No       | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description       | Schema   |
|----------------|-------------------|--|
| <b>200</b>     | Event/s requested | ⇔ <pre>▼ [   array of events   ► EventModel { } ]</pre>                  |
| <b>default</b> | Unexpected error  | ⇔ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

### Security

Security Schema

teamme\_application

Scopes

**read\_event**

## Summary

Insert a document

## Description

Insert a document with message, the response return ok and id of inserted document. The request must be a multipart/form-data

## Parameters

| Name         | Located in | Description   | Required | Schema    |
|--------------|------------|---|----------|-----------|
| payload.json | formData   | Payload JSON with all data for creation of a post. Must be a Json encoded string based on ItemNewDocument model | Yes      | ⇔ string  |
| impersonate  | formData   | id of user impersonate by application (require scope impersonate_any)   | No       | ⇔ integer |
| fileattach   | formData   | File attached at post   | Yes      | ⇔ file    |

## Responses

| Code           | Description       | Schema   |
|----------------|-------------------|--|
| <b>200</b>     | Inserted document | ⇔ <pre>▼ Insert {   result: ▶ string   id:    ▶ integer }</pre>                    |
| <b>default</b> | Unexpected error  | ⇔ <pre>▼ Error {   error:           ▶ string   error_description: ▶ string }</pre> |

## Security

| Security Schema    | Scopes              |
|--------------------|---------------------|
| teamme_application | <b>add_document</b> |

## Summary

Delete a document

## Description

Delete a document identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema                            |
|-------------|------------|---|----------|-----------------------------------|
| id_document | query      | id of document to delete  | Yes      | $\Leftrightarrow$ integer (int32) |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer         |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Document deleted | $\Leftrightarrow$ <pre>▼ Defaultmessage {   result: ► string }</pre>                     |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

## Security

| Security Schema    | Scopes                 |
|--------------------|------------------------|
| teamme_application | <b>delete_document</b> |

## Summary

Modify a document

## Description

Modify a document identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema  |
|-------------|------------|---|----------|---|
| body        | body       | Payload with post model to modify                                     | Yes      | <pre> ▼ DocumentModify {   id_document: ► integer   id_group: ► integer   isPublic: ► boolean   toFriends: ► boolean   message: ► string   user_list: ► [] } </pre> |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | <pre> ⇒ integer </pre>  |

## Responses

| Code           | Description       | Schema   |
|----------------|-------------------|--|
| <b>200</b>     | Document modified | <pre> ⇒ ▼ Defaultmessage {   result: ► string } </pre>                     |
| <b>default</b> | Unexpected error  | <pre> ⇒ ▼ Error {   error: ► string   error_description: ► string } </pre> |

## Security

| Security Schema    | Scopes                 |
|--------------------|------------------------|
| teamme_application | <b>modify_document</b> |

## Summary

Get one or more documents

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | query      | get documents in this group   | No       | ⇔ integer |
| id_document | query      | get document by id  | No       | ⇔ integer |
| versionable | query      | if true get only versionable documents otherwise also get attachments | No       | ⇔ boolean |
| date_from   | query      | get documents before this unixtimestamp (if 0 get lasts)              | No       | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description          | Schema   |
|----------------|----------------------|--|
| <b>200</b>     | Document/s requested | ⇔ <pre>▼ [   array of documents   ► DocumentModel { } ]</pre>            |
| <b>default</b> | Unexpected error     | ⇔ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

## Security

| Security Schema    | Scopes               |
|--------------------|----------------------|
| teamme_application | <b>read_document</b> |

## Summary

Insert a Task

## Description

Insert a task with attachments, the response return ok and id of inserted event. The request must be a multipart/form-data

## Parameters

| Name         | Located in | Description   | Required | Schema    |
|--------------|------------|---|----------|-----------|
| payload.json | formData   | Payload JSON with all data for creation of an event. Must be a Json encoded string based on ItemNewTask model | Yes      | ⇔ string  |
| impersonate  | formData   | id of user impersonate by application (require scope impersonate_any)   | No       | ⇔ integer |
| fileattach   | formData   | File attached at task   | No       | ⇔ file    |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Inserted task    | ⇔ <pre>▼ Insert {   result: ▶ string   id:    ▶ integer }</pre>                    |
| <b>default</b> | Unexpected error | ⇔ <pre>▼ Error {   error:           ▶ string   error_description: ▶ string }</pre> |

## Security

| Security Schema    | Scopes          |
|--------------------|-----------------|
| teamme_application | <b>add_task</b> |

## Summary

Delete a task

## Description

Delete a task identified by its id

## Parameters

| Name        | Located in | Description   | Required | Schema                            |
|-------------|------------|---|----------|-----------------------------------|
| id_task     | query      | id of task to delete  | Yes      | $\Leftrightarrow$ integer (int32) |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer         |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Task deleted     | $\Leftrightarrow$ <pre>▼ Defaultmessage {   result: ► string }</pre>                     |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

## Security

| Security Schema    | Scopes             |
|--------------------|--------------------|
| teamme_application | <b>delete_task</b> |

## Summary

Get one or more tasks

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | query      | get tasks in this group   | No       | ⇔ integer |
| id_task     | query      | get task by id  | No       | ⇔ integer |
| date_from   | query      | get tasks before this unixtimestamp (if 0 get lasts)                  | No       | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Task/s requested | ⇔ <pre>▼ [   array of tasks   ► TaskModel { } ]</pre>                    |
| <b>default</b> | Unexpected error | ⇔ <pre>▼ Error {   error: ► string   error_description: ► string }</pre> |

## Security

| Security Schema    | Scopes           |
|--------------------|------------------|
| teamme_application | <b>read_task</b> |

## Summary

Modify a task

## Description

Modify a task identified by its id

## Parameters

| Name              | Located in | Description   | Required | Schema  |
|-------------------|------------|---|----------|---|
| body              | body       | Payload with task model to modify                                     | Yes      | <pre> <b>TaskModify</b> {   id_task:   id_group:   isPublic:   toFriends:   message:   task_type:   assignee_user:   task_deadline:   task_startdate:   task_hours_assigned:   task_hours_spent:   task_budget_assigned:   task_progress:   user_list: } </pre> |
| impersonate query |            | id of user impersonate by application (require scope impersonate_any) | No       | ⇒ integer   |

## Responses

| Code | Description | Schema |
|------|-------------|--------|
|------|-------------|--------|

|                |                  |   |  |
|----------------|------------------|---|--|
| <b>200</b>     | Task modified    | ⇔ | <b>▼ Defaultmessage {</b><br>result: ▶ string<br><b>}</b>                      |
| <b>default</b> | Unexpected error | ⇔ | <b>▼ Error {</b><br>error: ▶ string<br>error_description: ▶ string<br><b>}</b> |

### Security

| Security Schema    | Scopes             |
|--------------------|--------------------|
| teamme_application | <b>modify_task</b> |

## Summary

Set progress on task

## Description

Modify progress on a task identified by its id

## Parameters

| Name              | Located in | Description   | Required | Schema  |
|-------------------|------------|---|----------|---|
| body              | body       | Payload with task progress model to modify                            | Yes      | $\Leftrightarrow$ <pre> TaskProgress {   id_task: inteq   task_hours_spent: inteq   task_money_spent: inteq   task_progress: inteq   message: string }                     </pre> |
| impersonate query |            | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer   |

## Responses

| Code           | Description             | Schema   |
|----------------|-------------------------|--|
| <b>200</b>     | Tasks progress modified | $\Leftrightarrow$ <pre> Defaultmessage {   result: string }                     </pre>                   |
| <b>default</b> | Unexpected error        | $\Leftrightarrow$ <pre> Error {   error: string   error_description: string }                     </pre> |

## Security

| Security Schema    | Scopes             |
|--------------------|--------------------|
| teamme_application | <b>modify_task</b> |

## Summary

Get a livestream

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | query      | get livestream of this group  | No       | ⇔ integer |
| id_user     | query      | get livestream of this user   | No       | ⇔ integer |
| date_from   | query      | get documents before this unixtimestamp (if 0 get lasts)              | No       | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code    | Description          | Schema   |
|---------|----------------------|--|
| 200     | Livestream requested | ⇔ <b>▼ LivestreamRequested</b> undefined<br><i>array of items ordered by last activity time (insert or modify time), an array of items for each moment</i><br><b>all of:</b> <ul style="list-style-type: none"> <li>▶ <b>PostsRequested</b> []</li> <li>▶ <b>EventsRequested</b> []</li> <li>▶ <b>DocumentsRequested</b> []</li> <li>▶ <b>TaskRequested</b> []</li> <li>▶ { }</li> </ul> |
| default | Unexpected error     | ⇔ <b>▼ Error</b> { <ul style="list-style-type: none"> <li>error: ▶ string</li> <li>error_description: ▶ string</li> </ul> }  |

## Security

Security Schema

teamme\_application

Scopes

**read\_livestream**

## Summary

Get user's friend list

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | List of friends  | ⇔ <b>▼ FriendList</b> [<br><i>array of short friend definition</i><br>▶ { }<br>] |
| <b>default</b> | Unexpected error | ⇔ <b>▼ Error</b> {<br>error: ▶ string<br>error_description: ▶ string<br>}        |

## POST /User.sendFriendRequest/

User

### Summary

Send connection request

### Parameters

| Name            | Located in | Description   | Required | Schema    |
|-----------------|------------|---|----------|-----------|
| id_friend       | formData   | User's id to send connection request                                  | No       | ⇔ integer |
| friend_username | formData   | User's username to send connection request (when do not have id)      | No       | ⇔ string  |
| message         | formData   | Message to send for connection request                                | No       | ⇔ string  |
| impersonate     | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

### Security

| Security Schema    | Scopes            |
|--------------------|-------------------|
| teamme_application | <b>add_friend</b> |

## POST /User.acceptFriendRequest/

User

### Summary

Accept connection request

### Parameters

| Name            | Located in | Description   | Required | Schema    |
|-----------------|------------|---|----------|-----------|
| id_friend       | formData   | User's id from accept connection request                              | No       | ⇔ integer |
| friend_username | formData   | User's username from accept connection request (when do not have id)  | No       | ⇔ string  |
| impersonate     | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

### Security

| Security Schema    | Scopes            |
|--------------------|-------------------|
| teamme_application | <b>add_friend</b> |

## POST /User.declineFriendRequest/

User

### Summary

Refuse connection request

### Parameters

| Name            | Located in | Description   | Required | Schema    |
|-----------------|------------|---|----------|-----------|
| id_friend       | formData   | User's id from refuse connection request                              | No       | ⇔ integer |
| friend_username | formData   | User's username from refuse connection request (when do not have id)  | No       | ⇔ string  |
| impersonate     | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

### Security

| Security Schema    | Scopes            |
|--------------------|-------------------|
| teamme_application | <b>add_friend</b> |

## Summary

Remove connection

## Parameters

| Name            | Located in | Description   | Required | Schema    |
|-----------------|------------|---|----------|-----------|
| id_friend       | formData   | User's id of connection   | No       | ⇔ integer |
| friend_username | formData   | User's username of connection   | No       | ⇔ string  |
| impersonate     | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Result           | ⇔ <pre>▼ MessageResult {   result: ▶ string   message: ▶ string }</pre>  |
| <b>default</b> | Unexpected error | ⇔ <pre>▼ Error {   error: ▶ string   error_description: ▶ string }</pre> |

## Security

| Security Schema    | Scopes               |
|--------------------|----------------------|
| teamme_application | <b>delete_friend</b> |

## POST /User.changePassword/

User

### Summary

Change user's password

### Parameters

| Name         | Located in | Description   | Required | Schema    |
|--------------|------------|---|----------|-----------|
| old_password | formData   | User's old password   | Yes      | ⇔ string  |
| new_password | formData   | User's new password   | Yes      | ⇔ string  |
| impersonate  | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

## POST /User.changeEmail/

User

### Summary

Change user's email

### Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| new_email   | formData   | User's new email  | Yes      | ⇔ string  |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

## POST /User.changeLanguage/

User

### Summary

Change user's GUI language

### Parameters

| Name        | Located in | Description   | Required | Schema                    |
|-------------|------------|---|----------|---------------------------|
| language    | formData   | User's new language   | Yes      | $\Leftrightarrow$ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer |

### Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Result           | $\Leftrightarrow$ <pre>▼ MessageResult {   result: ▶ string   message: ▶ string }</pre>  |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <pre>▼ Error {   error: ▶ string   error_description: ▶ string }</pre> |

## POST /User.changeNameSurname/

User

### Summary

Change user's Name and Surname

### Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| name        | formData   | User's new Name   | Yes      | ⇔ string  |
| surname     | formData   | User's new Surname  | Yes      | ⇔ string  |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

## Summary

Set Email notifications preferences

## Parameters

| Name | Located in | Description                               | Required | Schema   |
|------|------------|---|----------|--|
| body | body       | User's Email notification settings object | Yes      | <pre>                     ▼ EmailNotifications {                       email_notify: ▶ bo                       *                       email_digest_type: ▶ st                       *                       email_digest_notify_friends: ▶ bo                       email_digest_notify_private: ▶ bo                       email_digest_notify_content_new: ▶ bo                       email_digest_group_deny_list: ▶                     }                 </pre> |

## Responses

| Code    | Description      | Schema   |
|---------|------------------|--|
| 200     | Result           | <pre>                     ▼ MessageResult {                       result: ▶ string                       message: ▶ string                     }                 </pre>  |
| default | Unexpected error | <pre>                     ▼ Error {                       error: ▶ string                       error_description: ▶ string                     }                 </pre> |

## Summary

Get user's groups list

## Parameters

| Name        | Located in | Description   | Required | Schema                    |
|-------------|------------|---|----------|---------------------------|
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | $\Leftrightarrow$ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | List of group    | $\Leftrightarrow$ <code>▼ GroupList[<br/>array of short group definition<br/>▶ { }<br/>]</code>    |
| <b>default</b> | Unexpected error | $\Leftrightarrow$ <code>▼ Error {<br/>error: ▶ string<br/>error_description: ▶ string<br/>}</code> |

## Summary

Get group categories

## Parameters

| Name              | Located in | Description   | Required | Schema                       |
|-------------------|------------|---|----------|------------------------------|
| impersonate_query |            | id of user impersonate by application (require scope impersonate_any) | No       | $\rightleftharpoons$ integer |

## Responses

| Code           | Description        | Schema   |
|----------------|--------------------|--|
| <b>200</b>     | List of categories | $\rightleftharpoons$ <pre>▼ <b>CategoryList</b>[<br/>  array of categories definition<br/>  ▶ { }<br/>]</pre>  |
| <b>default</b> | Unexpected error   | $\rightleftharpoons$ <pre>▼ <b>Error</b> {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

## Summary

add new group

## Parameters

| Name        | Located in | Description   | Required | Schema  |
|-------------|------------|---|----------|---|
| body        | body       | New group object  | Yes      | <pre>           ▼ NewGroup {             privacy: ▶ string *             title: ▶ string *             description: ▶ string             category: ▶ integer *           }         </pre> |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | integer   |

## Responses

| Code    | Description       | Schema  |
|---------|-------------------|---|
| 200     | New Group success | <pre>           ▼ {             result: ▶ string             message: ▶ string             id_group: ▶ integer             url: ▶ string           }         </pre> |
| default | Unexpected error  | <pre>           ▼ Error {             error: ▶ string             error_description: ▶ string           }         </pre>  |

## Security

| Security Schema    | Scopes           |
|--------------------|------------------|
| teamme_application | <b>add_group</b> |

## Summary

modify group

## Parameters

| Name        | Located in | Description   | Required | Schema  |
|-------------|------------|---|----------|---|
| body        | body       | Modify group object   | Yes      | <pre>                     ▼ <b>ModifyGroup</b> {                       id_group: ▶ integer *                       privacy: ▶ string *                       title: ▶ string *                       description: ▶ string                       category: ▶ integer *                     }                 </pre> |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | <pre>                     ⇔ integer                 </pre>  |

## Responses

| Code    | Description          | Schema  |
|---------|----------------------|---|
| 200     | Modify Group success | <pre>                     ⇔ {                       result: ▶ string                       message: ▶ string                       id_group: ▶ integer                       url: ▶ string                     }                 </pre> |
| default | Unexpected error     | <pre>                     ⇔ ▼ <b>Error</b> {                       error: ▶ string                       error_description: ▶ string                     }                 </pre>   |

## Security

| Security Schema    | Scopes              |
|--------------------|---------------------|
| teamme_application | <b>modify_group</b> |

## DELETE /Group/

Group

### Summary

delete group

### Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | formData   | Group's id to delete  | Yes      | ⇔ integer |
| impersonate | query      | id of user impersonate by application (require scope impersonate_any) | No       | ⇔ integer |

### Responses

| Code           | Description      | Schema  |
|----------------|------------------|---|
| <b>200</b>     | Result           | ⇔<br><pre>▼ MessageResult {<br/>  result: ▶ string<br/>  message: ▶ string<br/>}</pre>  |
| <b>default</b> | Unexpected error | ⇔<br><pre>▼ Error {<br/>  error: ▶ string<br/>  error_description: ▶ string<br/>}</pre> |

### Security

| Security Schema    | Scopes              |
|--------------------|---------------------|
| teamme_application | <b>delete_group</b> |

## Parameters

| Name              | Located in | Description   | Required | Schema   |
|-------------------|------------|---|----------|--|
| body              | body       | New group's user object   | Yes      | <pre> ↔ ▼ NewGroupUser {   id_group: ▶ integer *   id_user: ▶ integer   username: ▶ string   ACL: ▶ integer } </pre> |
| impersonate query |            | id of user impersonate by application (require scope impersonate_any) | No       | ↔ integer  |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Result           | <pre> ↔ ▼ MessageResult {   result: ▶ string   message: ▶ string } </pre>  |
| <b>default</b> | Unexpected error | <pre> ↔ ▼ Error {   error: ▶ string   error_description: ▶ string } </pre> |

## Security

| Security Schema    | Scopes                    |
|--------------------|---------------------------|
| teamme_application | <b>manage_group_users</b> |

## Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | formData   | id of group   | Yes      | ⇔ integer |
| id_user     | formData   | id of user to remove  | No       | ⇔ integer |
| username    | formData   | username of user to remove<br>(when do not have id)                         | No       | ⇔ string  |
| impersonate | query      | id of user impersonate by<br>application (require scope<br>impersonate_any) | No       | ⇔ integer |

## Responses

| Code           | Description      | Schema   |
|----------------|------------------|--|
| <b>200</b>     | Result           | ⇔ <pre>           ▼ MessageResult {             result: ▶ string             message: ▶ string           }           </pre>  |
| <b>default</b> | Unexpected error | ⇔ <pre>           ▼ Error {             error: ▶ string             error_description: ▶ string           }           </pre> |

## Security

| Security Schema    | Scopes                    |
|--------------------|---------------------------|
| teamme_application | <b>manage_group_users</b> |

## POST /Group.modifyGroupMemberACL/

Group

### Parameters

| Name        | Located in | Description   | Required | Schema    |
|-------------|------------|---|----------|-----------|
| id_group    | formData   | id of group   | Yes      | ⇔ integer |
| id_user     | formData   | id of user to change  | No       | ⇔ integer |
| username    | formData   | username of user to change<br>(when do not have id)                                   | No       | ⇔ string  |
| ACL         | formData   | ACL assigned (0 only read, 1<br>read and reply, 2 read and write,<br>3 administrator) | Yes      | ⇔ integer |
| impersonate | query      | id of user impersonate by<br>application (require scope<br>impersonate_any)           | No       | ⇔ integer |

### Responses

| Code    | Description      | Schema  |
|---------|------------------|---|
| 200     | Result           | ⇔<br><pre>▼ MessageResult {   result: ▶ string   message: ▶ string }</pre>  |
| default | Unexpected error | ⇔<br><pre>▼ Error {   error: ▶ string   error_description: ▶ string }</pre> |

### Security

| Security Schema    | Scopes                    |
|--------------------|---------------------------|
| teamme_application | <b>manage_group_users</b> |

## Models

### Insert

```
⇔  
▼ Insert {  
  result: ▶ string  
  id: ▶ integer  
}
```

## Defaultmessage

```
▼ Defaultmessage {  
⇒  result: ▶ string  
}
```

## MessageResult

```
▼ MessageResult {  
⇒  result: ▶ string  
   message: ▶ string  
}
```

## Error

```
▼ Error {  
⇒  error: ▶ string  
   error_description: ▶ string  
}
```

## ItemNew

```
▼ ItemNew {  
   id_group: ▶ integer  
   isPublic: ▶ boolean  
⇒  toFriends: ▶ boolean  
   message: ▶ string  
   user_list: ▶ []  
}
```

## ItemNewEvent

```
▼ ItemNewEvent {  
   id_group: ▶ integer  
   isPublic: ▶ boolean  
   toFriends: ▶ boolean  
⇒  message: ▶ string  
   event_from: ▶ integer  
   event_to: ▶ integer  
   webEx: ▶ boolean  
   user_list: ▶ []  
}
```

## ItemNewDocument

```
▼ ItemNewDocument {  
   id_group: ▶ integer  
   id_version: ▶ integer  
⇒  isPublic: ▶ boolean  
   toFriends: ▶ boolean
```

```
    message: ▶ string
    user_list: ▶ []
}
```

## ItemNewTask

```
▼ ItemNewTask {
  id_group: ▶ integer
  isPublic: ▶ boolean
  toFriends: ▶ boolean
  message: ▶ string
  task_type: ▶ integer
⇒ assignee_user: ▶ integer
  task_deadline: ▶ integer
  task_startdate: ▶ integer
  task_hours_assigned: ▶ integer
  task_budget_assigned: ▶ integer
  user_list: ▶ []
}
```

## PostModify

```
▼ PostModify {
  id_post: ▶ integer
  id_group: ▶ integer
⇒ isPublic: ▶ boolean
  toFriends: ▶ boolean
  message: ▶ string
  user_list: ▶ []
}
```

## EventModify

```
▼ EventModify {
  id_event: ▶ integer
  id_group: ▶ integer
  isPublic: ▶ boolean
  toFriends: ▶ boolean
⇒ message: ▶ string
  event_from: ▶ integer
  event_to: ▶ integer
  webEx: ▶ boolean
  user_list: ▶ []
}
```

## DocumentModify

```

▼ DocumentModify {
  id_document: ▶ integer
  id_group:    ▶ integer
⇒ isPublic:   ▶ boolean
  toFriends:  ▶ boolean
  message:    ▶ string
  user_list:  ▶ []
}

```

## TaskModify

```

▼ TaskModify {
  id_task:           ▶ integer
  id_group:          ▶ integer
  isPublic:          ▶ boolean
  toFriends:         ▶ boolean
  message:           ▶ string
  task_type:         ▶ integer
⇒ assignee_user:    ▶ integer
  task_deadline:     ▶ integer
  task_startdate:   ▶ integer
  task_hours_assigned: ▶ integer
  task_hours_spent: ▶ integer
  task_budget_assigned: ▶ integer
  task_progress:    ▶ integer
  user_list:        ▶ []
}

```

## TaskProgress

```

▼ TaskProgress {
  id_task:           ▶ integer
  task_hours_spent: ▶ integer
⇒ task_money_spent: ▶ integer
  task_progress:    ▶ integer
  message:          ▶ string
}

```

## AttachmentModel

```

▼ AttachmentModel {
  id_document: ▶ integer
  file_name:   ▶ string
  file_size:   ▶ integer
⇒ file_time:  ▶ integer
  FileURL:    ▶ string
}

```

```
    PreviewURL: ▶ string
    isPublic:   ▶ boolean
}
```

## MessageModel

```
▼ MessageModel {
  title: ▶ string
  body:  ▶ string
}
```

## UserModel

```
▼ UserModel {
  id_user:           ▶ integer
  id_language:      ▶ integer
  user_name:        ▶ string
  user_surname:     ▶ string
  lastTimeOnline:  ▶ integer
  is_friend:        ▶ boolean
  pending_friend:  ▶ boolean
  onlineStatus:    ▶ string
  avatars:          ▶ []
  user_url:         ▶ string
  friends:          ▶ integer
}
```

## AvatarModel

```
▼ AvatarModel {
  file:   ▶ string
  md5:    ▶ string
  URL:    ▶ string
  fileMime: ▶ string
  ETag:   ▶ string
  filesize: ▶ integer
}
```

## PostModel

```
▼ PostModel {
  id_post:           ▶ integer
  post_time:         ▶ integer
  post_modification_date: ▶ integer
  reference_acl:     ▶ string
  reference_acl_id:  ▶ integer
  complete_acl:     ▶ boolean
}
```

```

⇒ itemRecipients:           ▶ []
postItemPermalink:         ▶ string
message_parts:             ▶ MessageModel { }
isFriend:                  ▶ boolean
isPublic:                  ▶ boolean
isGroup:                   ▶ boolean
postHasAttachments:        ▶ boolean
postAttachments:           ▶ []
user:                      ▶ UserModel { }
}

```

## EventModel

```

▼ EventModel {
  id_event:                 ▶ integer
  event_time:               ▶ integer
  event_modification_date: ▶ integer
  event_from:               ▶ integer
  event_to:                 ▶ integer
  webminary:                ▶ WebminaryModel { }
  reference_acl:            ▶ string
  reference_acl_id:         ▶ integer
  complete_acl:             ▶ boolean
⇒ itemRecipients:           ▶ []
eventItemPermalink:        ▶ string
message_parts:             ▶ MessageModel { }
isFriend:                  ▶ boolean
isPublic:                  ▶ boolean
isGroup:                   ▶ boolean
eventHasAttachments:       ▶ boolean
postAttachments:           ▶ []
user:                      ▶ UserModel { }
}

```

## DocumentModel

```

▼ DocumentModel {
  id_document:             ▶ integer
  file_time:               ▶ integer
  file_name:               ▶ string
  file_size:               ▶ integer
  id_fk:                   ▶ integer
  id_tool:                 ▶ integer
  version_revision:       ▶ integer
}

```

```

    last_revision: ▶ boolean
    reference_acl: ▶ string
    reference_acl_id: ▶ integer
    complete_acl: ▶ boolean
    itemRecipients: ▶ []
    message_parts: ▶ MessageModel { }
    isFriend: ▶ boolean
    isPublic: ▶ boolean
    isGroup: ▶ boolean
    attach_data: ▶ AttachmentModel { }
    user: ▶ UserModel { }
}

```

## TaskModel

```

▼ TaskModel {
    id_task: ▶ integer
    task_type: ▶ integer
    task_time: ▶ integer
    task_modification_date: ▶ integer
    task_deadline: ▶ integer
    task_hours_assigned: ▶ integer
    task_hours_spent: ▶ integer
    task_progress: ▶ integer
    reference_acl: ▶ string
    reference_acl_id: ▶ integer
    complete_acl: ▶ boolean
    itemRecipients: ▶ []
    message_parts: ▶ MessageModel { }
    isFriend: ▶ boolean
    isPublic: ▶ boolean
    isGroup: ▶ boolean
    attach_data: ▶ AttachmentModel { }
    user: ▶ UserModel { }
    assignee: ▶ UserModel { }
}

```

## WebminaryModel

```

▼ WebminaryModel {
    type: ▶ string
    session_key: ▶ string
    guest_token: ▶ string
    event_url: ▶ string
}

```

```
    event_passwd: ▶ string
  }
```

## PostsRequested

```
  ▼ PostsRequested[
⇒   array of posts
    ▶ PostModel { }
  ]
```

## EventsRequested

```
  ▼ EventsRequested[
⇒   array of events
    ▶ EventModel { }
  ]
```

## DocumentsRequested

```
  ▼ DocumentsRequested[
⇒   array of documents
    ▶ DocumentModel { }
  ]
```

## TaskRequested

```
  ▼ TaskRequested[
⇒   array of tasks
    ▶ DocumentModel { }
  ]
```

## LivestreamRequested

```
  ▼ LivestreamRequested undefined
    array of items ordered by last activity time (insert or modify time),
    an array of items for each moment
  all of:
⇒   ▶ PostsRequested[]
    ▶ EventsRequested[]
    ▶ DocumentsRequested[]
    ▶ TaskRequested[]
    ▶ { }
```

## FriendList

```
  ▼ FriendList[
⇒   array of short friend definition
    ▶ { }
  ]
```

## GroupList

```
  ▼ GroupList[
⇒   array of short group definition
    ▶ { }
  ]
```

## CategoryList

```
▼ CategoryList[  
⇒   array of categories definition  
    ▶ { }  
]
```

## EmailNotifications

```
▼ EmailNotifications {  
    email_notify: ▶ boolean *  
    email_digest_type: ▶ string *  
⇒   email_digest_notify_friends: ▶ boolean  
    email_digest_notify_private: ▶ boolean  
    email_digest_notify_content_new: ▶ boolean  
    email_digest_group_deny_list: ▶ []  
}
```

## NewGroup

```
▼ NewGroup {  
    privacy: ▶ string *  
⇒   title: ▶ string *  
    description: ▶ string  
    category: ▶ integer *  
}
```

## ModifyGroup

```
▼ ModifyGroup {  
    id_group: ▶ integer *  
    privacy: ▶ string *  
⇒   title: ▶ string *  
    description: ▶ string  
    category: ▶ integer *  
}
```

## NewGroupUser

```
▼ NewGroupUser {  
    id_group: ▶ integer *  
⇒   id_user: ▶ integer  
    username: ▶ string  
    ACL: ▶ integer  
}
```