

BUDEMI API DOCUMENTATION

Deployment: <http://3.89.218.253:8000/>

Register

POST /app/register/

Description

Registers the given user to Budemi. Returns a session token for the registered user. This token is necessary to use user-specific API calls.

Parameters

Parameter	Type	Description
username	varchar(200) [required]	username of the registered user
email	varchar(200) [required]	email of the registered user
password	varchar(200) [required]	password of the registered user

Responses

Response Code	Example Response Bodies
200 (OK)	<pre>{ "user": { "id": 1, "username": "example_username", "email": "example@mail.com" }, "token": "de58kldjf...sdfsa" }</pre>
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None

Login

POST /app/login/

Description

Users can log in to Budemi with their username and password if they already have a Budemi account with the given credentials. Returns a session token.

Parameters

Parameter	Type	Description
username	varchar(200) [required]	username of the user
password	varchar(200) [required]	password of the user

Responses

Response Code	Example Response Bodies
200 (OK)	{ "expiry" "2022-10-29T20:18:50.861039Z": "token": "de58kldjf...sdfsa" }
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None

Logout

POST /app/logout/

Description

User can log out. This is done by closing the given session (token). The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

None

Responses

Response Code	Example Response Bodies
200 (OK)	None
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None

Change Password

PUT /app/change-password/

Description

Users can change their passwords using this endpoint. The old and new passwords of the user should be specified. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
-----------	------	-------------

old_pass	varchar(200) [required]	old password of the user
new_pass	varchar(200) [required]	new password of the user

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "status": "success", "code": 200, "message": "Password updated successfully" }</pre>
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None

Learning Space

Description

Users can create a new learning space. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

POST /app/learning-space/

Parameters

Parameter	Type	Description
name	varchar(30) [required]	Name of the learning space
tag	varchar(30) [required]	Tag of the learning space
description	TextField	Description of the learning space

image	ImageField	This field is optional
-------	------------	------------------------

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 4, "name": "Django", "members": [{ "id": 5, "username": "kir8", "email": "kadrka@hotmail.com" }], "tag": "Tech", "image": "/media/uploads/r_cAfKZbL.jpg", "ls_owner": 5, "description": "Django into tutorial", "created_on": "2022-12-24T23:46:46.365747Z" }</pre>
400 (Bad request)	None

GET /app/learning-space/

Parameters

Parameter	Type	Description
id	int	id of the learning space

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 4, "name": "Django", "members": [{ "id": 5, "username": "kadir",</pre>

	<pre> "email": "kadrka@hotmail.com" },], "tag": "Tech", "image": "/media/uploads/r_cAfKZbL.jpg", "ls_owner": { "id": 5, "username": "kadir", "email": "kadrka@hotmail.com" }, "description": "Django into tutorial", "created_on": "2022-12-24T23:46:46.365747Z" } </pre>
400 (Bad request)	{"message": "given id doesn't exist"}

Content

Description

Users can add content to an existing learning space. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

POST /app/content/

Parameters

Parameter	Type	Description
name	varchar(30) [required]	Name of the content
type	varchar(30) [required]	Type of the content. Type can be either text, video, image, discussion, or meeting
text	TextField	Required if the “type” of the content is either text or meeting

url	TextField	Required if the “type” of the content is either video or image
learningSpace	int	Id of the learning space

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": "id of the content", "name": "name of the content", "type": "type of the content", "text": "text about content", "url": "url of the related content", "owner": "id of the owner ", "learningSpace": "id of the learning space ", "upVoteCount": "up vote count of the learning content ", }</pre>
400 (Bad request)	None

GET /app/content/

Parameters

Parameter	Type	Description
id	int	id of the content

Response

Response Code	Example Response Bodies
200 (OK)	<pre> } { "id": "id of the content", "name": "name of the content", "type": "type of the content", "text": "text about content", "url": "url of the related content", "owner": "id of the user ", </pre>

	<pre>" learningSpace ":" id of the learning space ", " upVoteCount ":" up vote count of the learning content ", }</pre>
400 (Bad request)	<pre>{"message": "given id doesn't exist"}</pre>

PATCH /app/content/

- This endpoint is used to update a content.

Parameters

Parameter	Type	Description
id	int [required]	id of the content
name	varchar(30)	Name of the content
type	varchar(30)	Type of the content. Type can be either text, video, image, or meeting
text	TextField	Required if the “type” of the content is either text or meeting
url	TextField	Required if the “type” of the content is either video or image
upVoteCount	int	Use it to update the upvote count

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 1, "name": "House Types", "type": "text", "text": "Good houses are many in Istanbul.", "url": "\"\"", "owner": 3, "learningSpace": 1, "upVoteCount": 1 }</pre>

	}
400 (Bad request)	None

Example Request in Postman:

Content List

Description

To get content in a learning space. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

GET /app/content-list/

Parameters

Parameter	Type	Description
learning_space_id	int	id of the learning space

Response

Response Code	Example Response Bodies
200 (OK)	<pre>[{ " id ":" id of the content ", " name ":" name of the content ", " type ":" type of the content ", " text ":" text about content ", " url ":" url of the related content ", " owner ":" id of the user ", " owner ":" id of the learning space ",</pre>

	<pre>" up VoteCount ":" up vote count of the learning content ", }]</pre>
400 (Bad request)	{"message": "given id doesn't exist"}

Enroll

POST /app/enroll/

Description

Users can enroll to a learning space and then add content. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
learning_space_id	int	Id of the learning space

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": "id of the learning space", "name": name of the learning space, "members": [{"id": id of the user, "username": "username of the user", "email": "email of the user" }] }</pre>
400 (Bad request)	{"message": "given learning space id doesn't exist"}

Discussion

POST /app/discussion/

Description

Users can discuss below a specific content. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
content	int	id of the content
body	text	the comment that user wants to do

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": id of the discussion entry, "content": id of the content, "owner": id of the owner of the entry, "body": "an example discussion entry here", "created_on": "2022-12-03T18:58:26.613305Z" }</pre>
400 (Bad request)	<pre>{"message": "given content id doesn't exist"}</pre>

Discussion List

GET /app/discussion-list/

Description

Retrieves all the discussions associated with specific content id. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
content_id	int	id of the content

Response

Response Code	Example Response Bodies
200 (OK)	<pre>[{ "id": 1, "content": 1, "owner": { "id": 2, "username": "ege", "email": "egetaga@gmail.com" }, "body": "I really liked this content here.", "created_on": "2022-12-03T18:58:26.613305Z" }]</pre>
400 (Bad request)	<pre>{"message": "given content id doesn't exist"}</pre>

Learning Space List

GET /app/learning-space-list/

Description

Lists all learning spaces in budemi. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 1, "name": "sport", "members": [{ "id": 1, "username": "kir4", "email": "kkadrkka@hotmail.com" }], "tag": "asdasd", "image": "/media/uploads/r_SCLBxKq.jpg", "ls_owner": { "id": 1, "username": "kir4", "email": "kkadrkka@hotmail.com" } }], }</pre>

	<pre> "description": "this is", "created_on": "2022-12-20T12:29:14.289015Z" }, { "id": 4, "name": "Django", "members": [{ "id": 5, "username": "kir8", "email": "kadrka@hotmail.com" }], "tag": "Tech", "image": "/media/uploads/r_cAfKZbL.jpg", "is_owner": { "id": 5, "username": "kir8", "email": "kadrka@hotmail.com" }, "description": "Django into tutorial", "created_on": "2022-12-24T23:46:46.365747Z" }] } </pre>
400 (Bad request)	{"message": "given id doesn't exist"}

Learning Space Search

GET /app/learning-space-search/

Description

Search in all learning spaces in budemi with a search parameter. We check whether the name of a learning space contains the search parameter. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
search_parameter	text	The keyword you search

Response

Response Code	Example Response Bodies
200 (OK)	<pre>[{ "id": 1, "name": "kurt", "members": [{ "id": 1, "username": "efe", "email": "efetaga@gmail.com" }], "tag": "animals" }]</pre>
400 (Bad request)	<pre>{"message": "given id doesn't exist"}</pre>

Learning Space Tag Search

GET /app/learning-space-tag-search/

Description

Search learning spaces that belong to a particular tag. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
tag	text	The tag you search

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 4, "name": "Django", "members": [{ "id": 5, "username": "kadir", "email": "kadrka@hotmail.com" }], "tag": "Tech", "image": "/media/uploads/r_cAfKZbL.jpg", "ls_owner": { "id": 5, "username": "kadir", "email": "kadrka@hotmail.com" }, "description": "Django into tutorial", }] }</pre>

	<pre> "created_on": "2022-12-24T23:46:46.365747Z" }] }</pre>
400 (Bad request)	{"message": "given id doesn't exist"}

Profile

Description

Users can create profile pages. The profile page includes user information and enrolled learning spaces. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

POST /app/profile/

Parameters

Parameter	Type	Description
about_me	varchar(30) [required]	Brief introduction of user
image	TextField [optional]	Profile image of the user

Response

Response Code	Example Response Bodies
200 (OK)	<pre> { "id": 1, "about_me": "Student", "user": 2, "learningspaces": [1], "image": "imagestring" }</pre>

400 (Bad request)	None
-------------------	------

GET /app/profile/

Parameters

Parameter	Type	Description
user_id	id [optional]	Id of the user (note: if user_id is not specified, id request sender user's id is used.

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 1, "about_me": "Student", "user": { "id": 2, "username": "kadir", "email": "kkadirkkalkan@gmail.com", "learningspaces": [1], "image": "imagestring" } }</pre>
400 (Bad request)	None

PATCH /app/profile/

- This endpoint is used to update the user profile.

Parameters

Parameter	Type	Description
about_me	CharField (100) [optional]	about_me field of profile model
image	TextField [optional]	image of the user

Response

Response Code	Example Response Bodies
---------------	-------------------------

200 (OK)	<pre>{ "id": 1, "about_me": "student", "user": 1, "image": "imagestring", "learningspaces": [] }</pre>
400 (Bad request)	<pre>{"error": "you are not the owner of this note"}</pre>

Reset Password

Description

A user can reset his/her password. POST method requires an email. Random created new email will be sent to the user, if a user exists with this email.

POST /app/forget-password/

Parameters

Parameter	Type	Description
email	varchar(30) [required]	email of the user

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{"message": "your password is sent to your email"}</pre>
400 (Bad request)	<pre>{"message": "given email doesn't exist"}</pre>

Enrolled Learning Space List

GET /app/enrolled-learning-spaces/

Description

Lists all learning spaces that a user has previously enrolled. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description

Response

Response Code	Example Response Bodies
200 (OK)	<pre>[{ "id": 1, "name": "kurt", "members": [{ "id": 1, "username": "efe", "email": "efetaga@gmail.com" }] },]</pre>

	<pre> "tag": "animals" }] </pre>
400 (Bad request)	{"message": "given id doesn't exist"}

leave Learning Space

POST /app/leave-learning-space/

Description

Users can leave a learning space that he/she has previously enrolled. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
learning_space_id	int	Id of the learning space

Response

Response Code	Example Response Bodies
200 (OK)	<pre> { "id": "id of the learning space", "name": name of the learning space, "members": [{"id": id of the user, "username": "username of the user", "email": "email of the user" }] } </pre>
400 (Bad request)	{"message": "given learning space id doesn't exist"}

User from ID

GET /app/user-from-id/

Description

Shows the user from its id.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
id	int [required]	user id

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }</pre>
400 (Bad request)	<pre>{"message": "User with given id doesn't exist"}</pre>

Favorite a Learning Space

GET /app/favorite/

Description

Returns the all favorite learning spaces of a user.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
user	int	user id

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 3, "user": 2, "learningSpace": { "id": 1, "name": "Learning to build your house", "members": [{ "id": 2,</pre>

	<pre> "username": "egetaga", "email": "egetaga@gmail.com" }], "tag": "Hobby", "image": "/media/uploads/types-of-homes-hero.jpeg", "is_owner": { "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }, "description": "In this learning space, let's explore collaboratively how to build your own house.", "created_on": "2022-12-23T17:21:00.874874Z" } }] } </pre>
400 (Bad request)	{"message": "given user id doesn't exist"}

Post /app/favorite/

Description

Adds a specific learning space to the favorite learning spaces of a user.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
learningSpace	int [required]	learningSpace id

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 3, "user": 2, "learningSpace": { "id": 1, "name": "Learning to build your house", "members": [{ "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }], "tag": "Hobby", "image": "/media/uploads/types-of-homes-hero.jpeg", "is_owner": { "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }, "description": "In this learning space, let's explore collaboratively how to build your own house.", "created_on": "2022-12-23T17:21:00.874874Z" } }] }</pre>
400 (Bad request)	<pre>{"message": "given user id doesn't exist"}</pre>

Unfavorite a Learning Space

Post /app/unfavorite/

Description

Unfavorites a learning space.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
learningSpace	int [required]	learningSpace id

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": null, "user": 2, "learningSpace": { "id": 1, "name": "Learning to build your house", "members": [{ "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }], "tag": "Hobby", "image": "/media/uploads/types-of-homes-hero.jpeg", "Is_owner": { "id": 2, "username": "egetaga", "email": "egetaga@gmail.com" }, "description": "In this learning space, let's explore collaboratively how to build your own house.", } }</pre>

	<pre> "created_on": "2022-12-23T17:21:00.874874Z" } } </pre>
400 (Bad request)	<pre> {"message": "given learning space id doesn't exist"} </pre>

NOTE

Description

Users can take note on the content page. The session token must be specified.

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

POST /app/note/

Parameters

Parameter	Type	Description
body	TextField	Body of the note
content	int	Content id

Response

Response Code	Example Response Bodies
---------------	-------------------------

200 (OK)	<pre>{ "id": 2, "content": 2, "owner": 2, "body": "it is my note", "created_on": "2022-12-24T12:09:46.382219Z" }</pre>
400 (Bad request)	None

GET /app/note/

Parameters

Parameter	Type	Description
content_id	int	id of the content

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 2, "content": 2, "owner": { "id": 2, "username": "kadir", "email": "kadrkkka@hotmail.com" }, "body": "it is my note", "created_on": "2022-12-24T12:09:46.382219Z" }, { "id": 3, "content": 2, "owner": {</pre>

	<pre> "id": 2, "username": "kadir", "email": "kadrkkka@hotmail.com" }, "body": "it is my second note", "created_on": "2022-12-24T12:37:43.706006Z" }] }</pre>
400 (Bad request)	{"message": "given id doesn't exist"}

PATCH /app/note/

- This endpoint is used to update a note.

Parameters

Parameter	Type	Description
id	int [required]	id of the note
body	TextField	New body of the note

Response

Response Code	Example Response Bodies
200 (OK)	<pre> { "id": 2, "content": 2, "owner": { "id": 2, "username": "kadir", "email": "kadrkkka@hotmail.com" }, "body": "it is not", "created_on": "2022-12-24T12:09:46.382219Z" }</pre>

400 (Bad request)	<pre>{"error": "you are not the owner of this note"}</pre>

User from Username

GET /app/user_id_from_username/

Description

Gets the user from given username

Authorization

Authorization parameters should be given as header

key: Authorization

value: Token <session token>

Parameters

Parameter	Type	Description
username	CharField [required]	username of a user

Response

Response Code	Example Response Bodies
200 (OK)	<pre>{ "id": 2, "username": "kadir", "email": "kadrkkka@hotmail.com" }</pre>
400 (Bad request)	<pre>{"message": "User with given username doesn't exist"}</pre>

BUDEMI Annotation

DOCUMENTATION

Deployment: <http://3.89.218.253:8001/>

Annotation

GET /app/annotation/

Description

Retrieves all the annotations of a given content_id.

Parameters

Parameter	Type	Description
content_id	int [required]	id of the content

Responses

Response Code	Example Response Bodies
200 (OK)	<pre>{ "data": [{ "id": 2, "content_id": 1, "annotation_string": "{\ "@context\":"http://www.w3.org/ns/anno.jsonld",\ "type\":" "\Annotation",\ "body\":[{\ "type\":"TextualBody",\ "value\":" "\an old man",\ "purpose\":"commenting",\ "creator\":{\ "id\":"08fb2d b0408d7908b16696c2147001bbf5943ff98b97db6172788311b9ff323b", \ "name\":"mustafa"},\ "created\":"2022-12-26T12:29:58.343Z\ ",\ "modified\":"2022-12-26T12:29:59.367Z"}],\ "target\":{\ "s elector\":[{\ "type\":"TextQuoteSelector",\ "exact\":"Puerto Ricans"},{\ "type\":"TextPositionSelector",\ "start\":124,\ end\":137}]],\ "id\":"#4b2da4ad-627a-43da-b0c5-17b7cdd4a8e6" }" }, {</pre>

	<pre> "id": 29, "content_id": 1, "annotation_string": "{\\"@context\\":\\"http://www.w3.org/ns/anno.jsonld\\",\\"type\\": \\"Annotation\\",\\"body\\":[{\\"type\\":\\"TextualBody\\",\\"value\\": \\"test\\",\\"purpose\\":\\"commenting\\",\\"creator\\":{\\"id\\":\\"1aa 29efead9a6d502a14a5e01096628c3c4ff9b8a469ac459fb8e6fb9a848114 \",\\"name\\":\\"asd1\\"},\\"created\\":\\"2022-12-26T16:35:18.052Z\\ \",\\"modified\\":\\"2022-12-26T16:35:18.652Z\\"}],\\"target\\":{\\"s elector\\":[{\\"type\\":\\"TextQuoteSelector\\",\\"exact\\":\\"st\\"}, {\\"type\\":\\"TextPositionSelector\\",\\"start\\":7,\\"end\\":9}],\ \\"id\\":\\"#19392377-5477-4d28-a826-d411cbc791ef\\"}" </pre>
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None
400 (Bad request)	None

POST /app/annotation/

Description

Posts a given annotation.

Parameters

Parameter	Type	Description
content_id	int [required]	id of the content
annotation_string	Json String [required]	annotation string in the json form

Response

Response Code	Example Response Bodies
---------------	-------------------------

201 (Created)	<pre>{ "id": 46, "content_id": 10000, "annotation_string": "{\n \"@context\": \"http://www.w3.org/ns/anno.jsonld\", \n \"id\": \"http://example.org/anno20\", \n \"type\": \"Annotation\", \n \"body\": {\n \"type\" : \"TextualBody\", \n \"value\" : \"<p>example!</p>\", \n \"format\" : \"text/html\", \n \"language\" : \"en\" \n }, \n \"target\": {\n \"source\": \"http://example.org/post-slug\", \n\t\"selector\": {\n\t\t\"type\": \"FragmentSelector\", \n\t\t\"conformsTo\": \"http://www.w3.org/TR/media-frags/\", \n\t\t\"value\": \"xywh=50,50,640,480\" \n\t} \n} }</pre>
400 (Bad request)	None

PATCH /app/annotation/

Description

Alters a given annotation.

Parameters

Parameter	Type	Description
id	int [required]	id of the annotation
annotation_string	Json String [required]	annotation string in the json form

Response

Response Code	Example Response Bodies
200 (Ok)	<pre>{ "id": 46, "content_id": 10000, "annotation_string": "{\n \"@context\": \"http://www.w3.org/ns/anno.jsonld\", \n \"id\":</pre>

	<pre> {"http://example.org/anno20",\n "type": "Annotation",\n "body": {\n "type" : "TextualBody",\n "value" :\n "<p>example!</p>",\n "format" : "text/html",\n "language" : "en"\n },\n "target": {\n "source":\n "http://example.org/post-slug",\n "selector":\n {\n "type": "FragmentSelector",\n "conformsTo":\n "http://www.w3.org/TR/media-frags/",\n "value":\n "xywh=50,1000,640,480"\n }\n }\n} </pre>
400 (Bad request)	None

Annotation

POST /app/delete-annotation/

Description

Deletes the annotation of a given id.

Parameters

Parameter	Type	Description
id	int [required]	id of the content

Responses

Response Code	Example Response Bodies
200 (OK)	<pre> { "id": null, "content_id": 10000, "annotation_string": "{\n "@context":\n "http://www.w3.org/ns/anno.jsonld",\n "id":\n "http://example.org/anno20",\n "type": "Annotation",\n "body": {\n "type" : "TextualBody",\n "value" : </pre>

	<pre> \"<p>example!</p>\", \n \"format\" : \"text/html\", \n \"language\" : \"en\" \n \n }, \n \"target\": { \n \"source\": \"http://example.org/post-slug\", \n\t\"selector\": { \n\t\"type\": \"FragmentSelector\", \n\t\"conformsTo\": \"http://www.w3.org/TR/media-frags/\", \n\t\"value\": \"xywh=50,1000,640,480\" \n\t} \n} } </pre>
250 (Bad request)	None
403 (Forbidden)	None
404 (Not found)	None
400 (Bad request)	None