



Backend API Documentation

1.0.0 OAS 3.0

CFTrainer

Servers

http://localhost:3000

Assignments



POST **/assignment/create** Create assignment



Creates a new assignment within a group. Authentication Admin/Coach

Parameters

No parameters

Request body **required**

application/json

Example Value Schema

```
{
  "title": "string",
  "description": "string",
  "parent_group": "string",
  "due_date": "2026-01-27T07:18:55.691Z"
}
```

Responses

Code	Description
201	Assignment created successfully

Code	Description
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.692Z", "parent_group": "string"}</pre></div>
400	<div>Bad request - Invalid group ID or missing required fields</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

Retrieves assignments matching optional filter criteria. All filters are optional and can be combined. Authentication Basic



Parameters

Name	Description
title string (query)	Filter by assignment title (partial match) <div>title</div>
description string (query)	Filter by description (partial match) <div>description</div>
parent_group string (query)	Filter by parent group ID <div>parent_group</div>
due_date string(\$date-time) (query)	Filter by due date <div>due_date</div>



Responses

Code	Description
200	<div>Array of matching assignments</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "_id": "string", "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.702Z", "parent_group": "string" }]</div>
403	<div>Unauthorized</div> <div>Media type</div>

Code	Description
	<div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	Server error <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

GET	/assignment/get/{id}	Get assignment by ID	 
Retrieves a specific assignment by its ID. Authentication Basic			
Parameters			
Name	Description		
id * required string (path)	Assignment ID (must be valid MongoDB ObjectId) <div>id</div>		
Responses			
Code	Description		
200	Assignment found <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div>		

Code	Description
	<pre>{ "_id": "string", "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.704Z", "parent_group": "string" }</pre>
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Assignment not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

PUT	/assignment/update/{id}	Update assignment	 
Updates an existing assignment. Authentication Admin/Coach			
Parameters			
Name	Description		

Name	Description
id ★ required string (path)	Assignment ID (must be valid MongoDB ObjectId) <div>id</div>
Request body	<div>application/json</div>

Example Value	Schema
<pre>{ "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.708Z" }</pre>	

Responses



Code	Description
200	Assignment updated successfully <div>Media type <div>application/json</div> Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "_id": "string", "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.708Z", "parent_group": "string" }</pre>
403	Unauthorized <div>Media type <div>application/json</div></div> <div>Example Value Schema</div> <pre>{ "message": "string" }</pre>
404	Assignment not found <div>Media type</div>

Code	Description
	<div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
500	Server error <div>Media type</div> <div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

DELETE

/assignment/delete/{id}

Delete assignment



Deletes an assignment and all associated student exercises. Authentication Admin/Coach

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>Assignment ID (must be valid MongoDB ObjectId)</div> <div>id</div>

Responses

Code	Description
200	Assignment deleted successfully
403	Unauthorized

Media type

application/json

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	Assignment not found
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	Server error
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Authentication

^

POST	/auth/login	Login user with credentials	^
Authenticates a user with username and password, returning a JWT token for subsequent requests. Authentication N/A			
Parameters			
No parameters			
Request body		required	application/json
<div>Example Value Schema</div> <div><pre>{ "username": "string", "password": "string" }</pre></div>			

}

Responses

Code	Description
200	<div>Login successful</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "token": "string" }</pre></div>
400	<div>Invalid username or password</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

POST **/auth/register** Register new user account



Creates a new user account. Students can optionally provide a CodeForces account handle. Admins cannot be registered through this endpoint. Authentication N/A

Parameters

No parameters

Request body required

application/json

Example Value Schema

```
{
  "username": "string",
  "password": "string",
  "role": "student",
  "email": "string",
  "cf_account": "string"
}
```

Responses

Code	Description
201	<div>User registered successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "token": "string" }</pre></div>
400	<div>Bad request - Invalid role, duplicate username, or invalid cf_account for non-students</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
	}

POST

/auth/refresh-token

Refresh expired JWT token



Generates a new JWT token using the current authentication session. Use this when the current token is about to expire. Authentication Basic

Parameters

No parameters


Responses

Code	Description
200	<div>Token refreshed successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "token": "string" }</div>
401	<div>Unauthorized - Invalid or missing token</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{</div>

Code	Description
	<pre>"message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

CodeForces Account

^



GET	/cfaccount/me	Get authenticated student's CodeForces account	 ^
Retrieves the CodeForces account information linked to the authenticated student. Authentication Student			
Parameters			
No parameters			
Responses			
Code	Description		
200	CodeForces account found		
	Media type		
	<div>application/json</div>		
	Controls Accept header.		
	Example Value Schema		
	<pre>{ "_id": "string", "student_id": "string", "cf_account": "string", "is_verified_flag": true }</pre>		

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET

/cfaccount/get

Get CodeForces accounts by filters





Retrieves CodeForces accounts matching optional filter criteria. Authentication Admin

Parameters

Name	Description
student_id string (query)	Filter by student ID <div>student_id</div>
cf_account string (query)	Filter by CodeForces username <div>cf_account</div>
is_verified_flag string (query)	Filter by verification status (true/false) <div>is_verified_flag</div>

Responses

Code	Description
200	<div>Array of matching CodeForces accounts</div> <div>Media type<div>application/json</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>[{ "_id": "string", "student_id": "string", "cf_account": "string", "is_verified_flag": true }]</pre></div>
403	<div>Unauthorized</div> <div>Media type<div>application/json</div></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type<div>application/json</div></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>

GET	/cfaccount/get/{id}	Get CodeForces account by ID	 
Retrieves a specific CodeForces account by its ID. Authentication Admin			
Parameters			
Name	Description		
id ★ required	CFAccount ID (must be valid MongoDB ObjectId)		

Name	Description
string <i>(path)</i>	<div>id</div>

Responses

Code	Description
200	<div>CodeForces account found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "cf_account": "string", "is_verified_flag": true}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>Not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>

GET

/cfaccount/start_verify

Start CodeForces account verification process



Initiates verification of a CodeForces account. Returns a verification token and code to submit. Authentication Student

Parameters

No parameters

Responses

Code	Description
200	<div>Verification process started</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "verification_token": "string", "cf_code": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

PUT

/cfaccount/end_verify/{verify_token}

Complete CodeForces account verification

Completes the verification process using the verification token received from start_verify endpoint. Authentication Student

Parameters

Name	Description
<div>verify_token * required</div> <div>string</div> <div>(path)</div>	<div>Verification token from start_verify endpoint</div> <div><div>verify_token</div></div>

Responses



Code	Description
200	<div>Account verified successfully</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "cf_account": "string", "is_verified_flag": true }</pre></div>
400	<div>Unauthorized</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>
401	Invalid token
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
403	Unauthorized
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

PUT

/cfaccount/update/{id}

Update CodeForces account handle



Updates the CodeForces username for a student's account. Students can only update their own, admins can update any. Authentication Admin/Student

Parameters

Name	Description
id ★ required	CFAccount ID (must be valid MongoDB ObjectId)
string	

Name	Description
(path)	<div>id</div>
Request body	
<div>application/json</div>	
Example Value	Schema
<pre>{ "cf_account": "string"}</pre>	


Responses

Code	Description
200	<div>CFAccount updated successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "cf_account": "string", "is_verified_flag": true}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>Not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value</div> <div>Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Challenges

^

POST	/challenge/create	Create challenge	 ^
Creates a new challenge for the authenticated student. Authentication Student			
Parameters			
No parameters			
Request body		required	<div>application/json</div>
Example Value Schema			
<pre>{ "cf_code": "string" }</pre>			
Responses			
Code	Description		
201	Challenge created successfully		
	Media type	<div>application/json</div>	

Code	Description
	<p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "_id": "string", "cf_code": "string", "student_id": "string", "is_completed_flag": true, "completion_type": "contest" }</pre>
400	<p>Bad request</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>
403	<p>Unauthorized</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>
500	<p>Server error</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>

POST

/challenge/create/{student_id}

Create challenge for student (admin)

Creates a new challenge for a specific student. Authentication Admin

Parameters	
Name	Description
student_id * required string (path)	<div>student_id</div>
Request body required	<div>application/json</div>
Example Value	Schema
<pre>{ "cf_code": "string" }</pre>	



Code	Description
201	<div>Challenge created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "_id": "string", "cf_code": "string", "student_id": "string", "is_completed_flag": true, "completion_type": "contest" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET

/challenge/get

Get challenges with filters



Retrieves challenges matching optional criteria. Students see only their own. Authentication Admin/Student

Parameters

Name	Description
student_id string (query)	<div>student_id</div>
cf_code string (query)	<div>cf_code</div>
is_completed_flag string (query)	<div>is_completed_flag</div>
completion_type string (query)	<div>completion_type</div>

Responses

Code	Description
200	<div>Array of challenges</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "_id": "string", "cf_code": "string", "student_id": "string", "is_completed_flag": true, "completion_type": "contest" }]</pre></div>
403	<div>Forbidden</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

GET /challenge/get/{id} Get challenge by ID



Retrieves a specific challenge by ID. Authentication Admin



Parameters

Name	Description
id * required string (path)	<div>id</div>

Responses

Code	Description
200	<div>Challenge found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "cf_code": "string", "student_id": "string", "is_completed_flag": true, "completion_type": "contest"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>Challenge not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

Code	Description
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>



DELETE	/challenge/delete/{id}	Delete challenge	 
Deletes a challenge. Students can only delete their own. Authentication Admin/Student			
Parameters			
Name	Description		
id * required string (path)	<div>id</div>		
Responses			
Code	Description		
200	Challenge deleted		
400	Bad request		
	Media type		
	<div>application/json</div>		
	Example Value Schema		
	<pre>{ "message": "string" }</pre>		
403	Unauthorized		
	Media type		

Code	Description
	<div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	Challenge not found
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	Server error
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

PUT

/challenge/verify/{id}

Verify challenge completion



Marks a challenge as completed. Authentication Admin/Student

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>id</div>

Request body

application/json

Example Value Schema

```
{
  "completion_type": "string"
}
```

Responses



Code	Description
200	<div>Challenge verified</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "_id": "string", "cf_code": "string", "student_id": "string", "is_completed_flag": true, "completion_type": "contest" }</div>
400	<div>Verification failed</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
404	<div>Challenge not found</div> <div>Media type</div>

Code	Description
	<div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

GET

/challenge/ask

Get random challenge problem



Returns a random CodeForces problem matching criteria. Authentication Student

Parameters

Name	Description
min_rating number (query)	<div>min_rating</div>
max_rating number (query)	<div>max_rating</div>
tags string (query)	<div>tags</div>

Responses

Code	Description
200	<div>Challenge problem found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "cf_code": "string", "name": "string", "rating": 0, "contestId": 0, "tags": ["string"]}</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>



Sends a direct message from the authenticated user to another user. Authentication Basic

Parameters

Name	Description
user_id * required string (path)	<div>user_id</div>

Request body required

application/json



Example Value Schema

```
{
  "message": "string"
}
```



Responses

Code	Description
201	<div>Message sent successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "_id": "string", "sender_id": "string", "receiver_id": "string", "message": "string", "createdAt": "2026-01-27T07:18:55.762Z" }</div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>
403	Blocked
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET	/direct-messages/conversation/	Get conversation partners	 
Retrieves list of user IDs that the authenticated user has conversation history with. Authentication Basic			
Parameters			
No parameters			
Responses			
Code	Description		
200	Array of user IDs		
	Media type <div>application/json</div>		
	Controls Accept header.		
	Example Value Schema		

Code	Description
	<pre>["string"]</pre>
403	Unauthorized Media type <div>application/json</div> Example Value Schema <pre>{ "message": "string" }</pre>
500	Server error Media type <div>application/json</div> Example Value Schema <pre>{ "message": "string" }</pre>

GET	/direct-messages/conversation/{user_id}	Get conversation messages	 
Retrieves all messages in conversation with specified user. Authentication Basic			
Parameters			
Name	Description		
user_id * required string (path)	<div>user_id</div>		
Responses			
Code	Description		

Code	Description
200	<div>Array of messages</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "_id": "string", "sender_id": "string", "receiver_id": "string", "message": "string", "createdAt": "2026-01-27T07:18:55.766Z" }]</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Adds a user to your block list. Authentication Basic

Parameters

Name	Description
user_id * required string (path)	<div>user_id</div>

Responses



Code	Description
201	<div>User blocked</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "user_id": "string", "blocked_user_id": "string" }</div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>

Code	Description
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

DELETE

/direct-messages/unblock/{user_id}

Unblock a user



Removes a user from your block list. Authentication Basic



Parameters

Name	Description
<div><div>user_id</div><div>* required</div><div>string</div><div>(path)</div></div>	<div>user_id</div>

Responses

Code	Description
200	User unblocked
400	User not in block list
	<div>Media type</div> <div>application/json</div>
	<div>Example Value</div> <div>Schema</div> <div><div>{</div><div>"message": "string"</div><div>}</div></div>
403	Unauthorized
	<div>Media type</div>


Code	Description
	<div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

GET	/direct-messages/blocked	Get blocked users	 
Retrieves list of all blocked user IDs. Authentication Basic			
Parameters			
No parameters			
Responses			
Code	Description		
200	Array of blocked user IDs		
	Media type		
	application/json		
	Controls Accept header.		
	Example ValueSchema		
	<pre>["string"]</pre>		

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Exercises

^

POST	/exercise/create	Create exercise	 ^
Creates a new exercise within an assignment. Authentication Admin/Coach			
Parameters			
No parameters			
Request body		required	<div>application/json</div>
Example Value Schema			
<pre>{ "name": "string", "cf_code": "string", "parent_assignment": "string" }</pre>			

Responses

Code	Description
201	<div>Exercise created successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "name": "string", "cf_code": "string", "parent_assignment": "string"}</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

Retrieves exercises matching optional filter criteria. All filters are optional. Authentication Basic



Parameters

Name	Description
name string (query)	Filter by exercise name (partial match) <div>name</div>
cf_code string (query)	Filter by CodeForces problem code <div>cf_code</div>
parent_assignment string (query)	Filter by parent assignment ID <div>parent_assignment</div>

Responses

Code	Description
200	<div>Array of matching exercises</div> <div>Media type<div>application/json</div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "_id": "string", "name": "string", "cf_code": "string", "parent_assignment": "string" }]</pre></div>
403	<div>Unauthorized</div> <div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
	}
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>



GET	/exercise/get/{id}	Get exercise by ID	 
Retrieves a specific exercise by its ID. Authentication Basic			
Parameters			
Name	Description		
id * required string (path)	Exercise ID (must be valid MongoDB ObjectId)		
	<div>id</div>		
Responses			
Code	Description		
200	Exercise found		
	Media type		
	<div>application/json</div>		
	Controls Accept header.		
	Example Value Schema		
	<pre>{ "_id": "string", "name": "string", "cf_code": "string", "parent_assignment": "string" }</pre>		

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Exercise not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

PUT

/exercise/update/{id}

Update exercise



Updates an existing exercise. Authentication Admin/Coach

Parameters

Name	Description
id <small>* required</small> string (path)	Exercise ID (must be valid MongoDB ObjectId) <div>id</div>

Request body

application/json

Example Value Schema

```
{
  "name": "string"
}
```

Responses



Code	Description
200	<div>Exercise updated successfully</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "name": "string", "cf_code": "string", "parent_assignment": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Exercise not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

DELETE

/exercise/delete/{id}

Delete exercise



Deletes an exercise. Associated student-exercise links may also be deleted. Authentication Admin/Coach

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>Exercise ID (must be valid MongoDB ObjectId)</div> <div>id</div>

Responses

Code	Description
200	Exercise deleted successfully
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><div>{</div><div>"message": "string"</div><div>}</div></div>
404	<div>Exercise not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><div>{</div></div>



Code	Description
	<pre>"message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Following



POST	/following/create	Create following relationship between two students	
Creates a following relationship where student_1 follows student_2. Admins can create followings for any student, but students can only create followings for themselves. Users cannot follow themselves. Authentication Admin/Student			
Parameters			
No parameters			
Request body		required	<div>application/json</div>
Example Value Schema			
<pre>{ "student_2_id": "string", "student_1_id": "string" }</pre>			
Responses			
Code	Description		
201	Following created successfully		
	Media type	<div>application/json</div>	

Code	Description
	<p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "_id": "string", "student_1_id": "string", "student_2_id": "string" }</pre>
400	<p>Bad request</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>
403	<p>Unauthorized</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>
500	<p>Server error</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>{ "message": "string" }</pre>

GET	/following/get Get followings by student IDs	 
<p>Retrieves following relationships filtered by student IDs. Students can only view their own followings (as student_1_id). Admins can view all followings with any filter combination. Authentication Admin/Student</p>		
<p>Parameters</p>		

Name	Description
student_1_id	Filter by follower ID
string (query)	<input type="text" value="student_1_id"/>
student_2_id	Filter by followed user ID
string (query)	<input type="text" value="student_2_id"/>

Responses

Code	Description
200	<div>Array of following relationships</div> <div>Media type</div> <div><input type="text" value="application/json"/></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>[{ "_id": "string", "student_1_id": "string", "student_2_id": "string" }]</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div><input type="text" value="application/json"/></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div><input type="text" value="application/json"/></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
------	-------------

GET

/following/get/{id}

Get following relationship by ID



Retrieves a specific following relationship by its ID. Authentication Admin

Parameters

Name	Description
id * required string (path)	Following relationship ID (must be valid MongoDB ObjectId) <div>id</div>

Responses



Code	Description
200	<div>Following relationship found</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div><div>{ "_id": "string", "student_1_id": "string", "student_2_id": "string" }</div></div>
403	<div>Unauthorized</div> <div>Media type</div> <div><div>application/json</div></div> <div><div>Example Value</div><div>Schema</div><div>{ "message": "string" }</div></div>
404	<div>Following not found</div> <div>Media type</div>

Code	Description
	<div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
500	Server error
	<div>Media type</div> <div>application/json</div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

DELETE

/following/delete/{id}

Delete following relationship



Removes a following relationship. Students can only delete their own followings. Admins can delete any following. Authentication Admin/Student

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>Following relationship ID (must be valid MongoDB ObjectId)</div> <div>id</div>

Responses

Code	Description
200	Following deleted successfully
403	Unauthorized

Media type



application/json

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Following not found</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

GET

/following/count/{user_id}

Get follower count for a student





Counts how many users are following the specified student (user_id). Authentication Basic

Parameters

Name	Description
user_id * required string (path)	Target student's ID (must be valid MongoDB ObjectId) <div><div>user_id</div></div>

Responses

Code	Description
200	<div><div>Follower count retrieved</div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div><div>Example Value</div><div>Schema</div></div><div><pre>{ "count": 0 }</pre></div></div>
400	<div><div>Invalid user_id format</div><div>Media type</div><div>application/json</div><div><div>Example Value</div><div>Schema</div></div><div><pre>{ "message": "string" }</pre></div></div>
403	<div><div>Unauthorized</div><div>Media type</div><div>application/json</div><div><div>Example Value</div><div>Schema</div></div><div><pre>{ "message": "string" }</pre></div></div>
500	<div><div>Server error</div><div>Media type</div><div>application/json</div><div><div>Example Value</div><div>Schema</div></div><div><pre>{ "message": "string" }</pre></div></div>

GET	/following	List students followed by authenticated user	 
Returns a list of usernames that the authenticated student is following. Authentication Student			

Parameters

No parameters

Responses

Code	Description
200	<div>List of followings</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "following": [{ "name": "string" }]}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

Groups



POST /group/create Create group



Creates a new study group. Authentication Admin/Coach

Parameters

No parameters

Request body required

application/json

Example Value Schema

```
{
  "name": "string",
  "description": "string",
  "parent_coach": "string"
}
```

Responses

Code	Description
------	-------------

201 Group created

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "_id": "string",
  "name": "string",
  "description": "string",
  "parent_coach": "string",
  "group_messages": [
    {
      "sender_id": "string",
      "message": "string",
      "timestamp": "2026-01-27T07:18:55.806Z"
    }
  ],
  "invite_code": "string"
}
```

400 Bad request

Media type

application/json



Example Value Schema

Code	Description
	<pre>{ "message": "string" }</pre>
403	Unauthorized
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET

/group/get

Get groups by filters



Retrieves groups matching optional criteria. Authentication Admin/Coach

Parameters

Name	Description
name string (query)	<div>name</div>
description string (query)	<div>description</div>
parent_coach string (query)	<div>parent_coach</div>

Responses

Code	Description
200	<div>Array of groups</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "_id": "string", "name": "string", "description": "string", "parent_coach": "string", "group_messages": [{ "sender_id": "string", "message": "string", "timestamp": "2026-01-27T07:18:55.810Z" }], "invite_code": "string" }]</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Retrieves a specific group by ID. Authentication Basic

Parameters

Name	Description
id ★ required string (path)	<div>id</div>

Responses



Code	Description
200	<div>Group found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "name": "string", "description": "string", "parent_coach": "string", "group_messages": [{ "sender_id": "string", "message": "string", "timestamp": "2026-01-27T07:18:55.813Z" }], "invite_code": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
404	Group not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

PUT

/group/update/{id}

Update group



Updates group information. Authentication Admin/Coach

Parameters

Name	Description
<div><div>id ★ required</div><div>string</div><div>(path)</div></div>	<div>id</div>

Request body

application/json

Example Value Schema

```
{  
  "name": "string",  
  "description": "string"  
}
```

Responses

Code	Description
200	<div>Group updated</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "name": "string", "description": "string", "parent_coach": "string", "group_messages": [{ "sender_id": "string", "message": "string", "timestamp": "2026-01-27T07:18:55.816Z" }], "invite_code": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Group not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{</pre></div>

Code	Description
	<pre>{ "message": "string" }</pre>

DELETE

/group/delete/{id}

Delete group



Deletes a group and all associated data. Authentication Admin/Coach

Parameters

Name	Description
id * required string (path)	<div>id</div>

Responses



Code	Description
200	Group deleted
403	Unauthorized <div>Media type<div>application/json</div><div>Example Value Schema<pre>{ "message": "string" }</pre></div></div>
404	Group not found <div>Media type<div>application/json</div><div>Example Value Schema<pre>{ "message": "string" }</pre></div></div>

Code	Description
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

POST

/group/create-invite-code/{id}

Create invite code



Generates a new invite code for group. Authentication Admin/Coach

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>id</div>

Responses



Code	Description
201	<div>Invite code created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><div>{</div><div>"invite_code": "string"</div><div>}</div></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Group not found</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

GET

/group/get-invite-code/{id}

Get invite code



Retrieves current invite code for group. Authentication Admin/Coach

Parameters

Name	Description
<div><div>id</div><div><div><div>★ required</div></div></div><div>string</div><div>(path)</div></div>	<div>id</div>

Responses

Code	Description
200	<div>Invite code found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "invite_code": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Group not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

DELETE	/group/delete-invite-code/{id}	Delete invite code	 
Removes invite code for group. Authentication Admin/Coach			

Parameters

Name	Description
------	-------------

id * required string (path)	
--	--

	<div>id</div>
--	---------------

Responses

Code	Description
------	-------------

200	Invite code deleted
-----	---------------------

403	Unauthorized
-----	--------------

Media type

	<div>application/json</div>
--	-----------------------------

Example Value Schema

```
{
  "message": "string"
}
```

404	Group not found
-----	-----------------

Media type

	<div>application/json</div>
--	-----------------------------

Example Value Schema

```
{
  "message": "string"
}
```

500	Server error
-----	--------------

Media type

	<div>application/json</div>
--	-----------------------------

Example Value Schema

```
{
  "message": "string"
}
```

Retrieves all messages in a group. Authentication Basic

Parameters

Name	Description
id ★ required string (path)	<div>id</div>

Responses



Code	Description
200	<div>Array of messages</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "sender_id": "string", "message": "string", "timestamp": "2026-01-27T07:18:55.830Z" }]</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
404	<div>Group not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

POST

/group/send-message/{id}

Send message to group



Posts a message to group's message board. Authentication Basic

Parameters

Name	Description
<div>id ★ required</div> <div>string</div> <div>(path)</div>	<div>id</div>

Request body required

application/json

Example Value Schema

```
{  
  "message": "string"  
}
```

Responses

Code	Description
201	Message posted

Code	Description
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "sender_id": "string", "message": "string", "timestamp": "2026-01-27T07:18:55.833Z" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Group not found</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
------	-------------

GET

/group/my-groups-summary

Get my groups summary



Retrieves summary of groups user is member of. Authentication Basic

Parameters

No parameters

Responses

Code	Description
200	<div>Groups summary</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "groupId": "string", "name": "string", "ownerUsername": "string", "members": 0, "dueAssignments": 0, "role": "member" }]</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div>

Code

Description

application/json

Example Value

Schema

```
{  
  "message": "string"  
}
```

GET

/group/details/{id}

Get group details

^

Retrieves detailed group information. Authentication Basic

Parameters

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>id</div>

Responses

Code

Description

200

Group details

Media type

application/json

Controls Accept header.

Example Value

Schema


```
{  
  "group": {  
    "_id": "string",  
    "name": "string",  
    "description": "string",  
    "owner": {  
      "_id": "string",  
      "username": "string"  
    }  
  },  
  "assignments": [  

```

Code	Description
	<pre>{ "_id": "string", "title": "string", "description": "string", "due_date": "2026-01-27T07:18:55.838Z", "exerciseCount": 0 }</pre>
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Group not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Statistics

^

GET	/stats/get-student-stats/{studentId}	Get student statistics	 ^
Retrieves aggregated statistics about a student's progress. Authentication Basic			

Parameters	
Name	Description
studentId * required string (path)	<div>studentId</div>


Responses

Code	Description
200	<div>Student statistics retrieved</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "user": { "userId": "string", "cfHandle": "string", "role": "string" }, "kpis": { "rating": 0, "solvedTotal": 0, "streakDays": 0 }, "ratingGraph": { "min": 0, "max": 0, "series": [{ "t": "2026-01-27", "rating": 0 }] }, "solvesByRating": { "binSize": 0, "bins": [</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{</pre></div>

Code	Description
	<pre> "message": "string" }</pre>
404	Student not found
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type <div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Student Exercises

^

POST	/student-exercise/create	Create student-exercise link	 ^
Creates a link between authenticated student and an exercise. Authentication Student			
Parameters			
No parameters			
Request body	required	<div>application/json</div>	
Example Value Schema			
<pre>{ "exercise_id": "string" }</pre>			

Responses

Code	Description
201	<div>Link created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "exercise_id": "string", "completion_type": "contest" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Creates a link for specific student. Authentication Admin

Parameters

Name	Description
student_id * required string (path)	<div>student_id</div>

Request body required	<div>application/json</div>
------------------------------	-----------------------------

Example Value Schema

```
{
  "exercise_id": "string"
}
```

Responses



Code	Description
201	<div>Link created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "exercise_id": "string", "completion_type": "contest" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET

/student-exercise/get

Get student-exercise links



Retrieves links matching filter criteria. Authentication Basic

Parameters

Name	Description
student_id string (query)	<div>student_id</div>
exercise_id string (query)	<div>exercise_id</div>
assignment_id string (query)	<div>assignment_id</div>
group_id string (query)	<div>group_id</div>

Name	Description
completion_type string (query)	<div>completion_type</div>

Responses



Code	Description
200	<div>Array of student-exercise links</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "_id": "string", "student_id": "string", "exercise_id": "string", "completion_type": "contest" }]</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div>

Code	Description
	<div>application/json</div>
	<div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

GET

/student-exercise/get/{id}

Get student-exercise link by ID



Retrieves a specific link. Authentication Admin

Parameters

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>id</div>

Responses



Code	Description
200	<div>Link found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example ValueSchema</div> <div><pre>{ "_id": "string", "student_id": "string", "exercise_id": "string", "completion_type": "contest" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Link not found</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div><div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

DELETE

/student-exercise/delete/{id}

Delete student-exercise link



Removes link between student and exercise. Authentication Admin

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>id</div>

Responses

Code	Description
200	Link deleted
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Link not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Student Groups



POST	/student-group/create	Create student-group membership	
Adds a student to a group. Authentication Admin/Coach			
Parameters			
No parameters			
Request body		required	<div>application/json</div>

Example Value Schema

```
{
  "group_id": "string",
  "student_id": "string"
}
```

Responses

Code	Description
201	<div>Membership created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "group_id": "string" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <pre>{ "message": "string" }</pre>

GET **/student-group/get** Get student-group links

Retrieves membership links matching filters. Authentication Basic



Parameters

Name	Description
group_id string (query)	<div>group_id</div>
student_id string (query)	<div>student_id</div>



Responses

Code	Description
200	<div>Array of membership links</div> <div>Media type</div> <div><div>application/json</div><div>Controls Accept header.</div></div> <div>Example Value Schema</div> <pre>[{ "_id": "string", "student_id": "string", "group_id": "string" }]</pre>

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET	/student-group/get/{id}	Get student-group link by ID	 
Retrieves a specific membership link. Authentication Admin			
Parameters			
Name	Description		
id * required string (path)	<div>id</div>		
Responses			
Code	Description		
200	Link found		
	Media type		
	<div>application/json</div>		
	Controls Accept header.		

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "_id": "string", "student_id": "string", "group_id": "string"}</pre></div>
403	Unauthorized <div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	Link not found <div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	Server error <div>Media type<div>application/json</div></div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

DELETE	/student-group/delete/{id}	Delete student from group	 
Removes a student from a group. Authentication Admin/Coach			
Parameters			
Name	Description		

Name	Description
id ★ required string (path)	<div>id</div>

Responses

Code	Description
200	Membership deleted
403	Unauthorized <div>Media type<div>application/json</div><div>Example Value Schema<pre>{ "message": "string" }</pre></div></div>
404	Link not found <div>Media type<div>application/json</div><div>Example Value Schema<pre>{ "message": "string" }</pre></div></div>
500	Server error <div>Media type<div>application/json</div><div>Example Value Schema<pre>{ "message": "string" }</pre></div></div>

Allows a student to join a group using an invite code. Authentication Student

Parameters

No parameters

Request body required

application/json

Example Value Schema

```
{
  "invite_code": "string"
}
```

Responses

Code	Description
201	<div>Successfully joined group</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "_id": "string", "student_id": "string", "group_id": "string" }</div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
404	<div>Group not found</div> <div>Media type<div>application/json</div></div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type<div>application/json</div></div> <div>Example ValueSchema</div> <div><pre>{ "message": "string" }</pre></div>

Admin Users



POST	/admin/create	Create user account	
Creates a new user with specified role. Authentication Admin			
Parameters			
No parameters			
Request body		required	<div>application/json</div>
<div>Example ValueSchema</div> <div><pre>{ "username": "string", "password": "string", "role": "string", }</pre></div>			

```
"email": "string",
"cf_account": "string"
}
```

Responses

Code	Description
201	<div>User created</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "username": "string", "password_hash": "string", "email": "string", "role": "student" }</pre></div>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div>

Code	Description
	<pre>{ "message": "string" }</pre>

GET

/admin/get

Get users by filters



Retrieves users with sensitive info. Authentication Admin

Parameters

Name	Description
username string (query)	<input type="text" value="username"/>
role string (query)	<input type="text" value="role"/>
email string (query)	<input type="text" value="email"/>

Responses



Code	Description
200	<div>Array of users with sensitive info</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>[{ "_id": "string", "username": "string", "password_hash": "string", "email": "string", "role": "student" }]</pre></div>

Code	Description
403]
	Unauthorized
	Media type
	<div>application/json</div>
500	Example Value Schema
	{ "message": "string" }
	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	{ "message": "string" }

GET

/admin/get/{id}

Get user by ID (sensitive info)



Retrieves user with sensitive information. Authentication Admin

Parameters

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>id</div>

Responses



Code	Description
200	User found
	Media type

Code	Description
	<div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "username": "string", "password_hash": "string", "email": "string", "role": "student"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>User not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>

PUT

/admin/update/{id}

Update user



Updates user details. Authentication Admin

Parameters

Name	Description
------	-------------

id * required string (path)	
--	--

	<div>id</div>
--	---------------

Request body

application/json

Example Value Schema

```
{
  "password": "string",
  "role": "string",
  "email": "string"
}
```

Responses

Code	Description
------	-------------

200	User updated
-----	--------------

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "_id": "string",
  "username": "string",
  "password_hash": "string",
  "email": "string",
  "role": "student"
}
```

400	Bad request
-----	-------------



Media type

application/json

Example Value Schema

```
{
  "message": "string"
}
```

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	User not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>



DELETE	/admin/delete/{id}	Delete user	 
Permanently deletes a user. Authentication Admin			
Parameters			
Name	Description		
id * required string (path)	<div>id</div>		

Responses	
Code	Description
200	User deleted
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	User not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET

/admin/get-by-username/{username}

Get user by username (sensitive)



Retrieves user by username with sensitive info. Authentication Admin

Parameters

Name	Description
------	-------------

Name	Description
username ★ required string (path)	<div>username</div>

Responses

Code	Description
200	<div>User found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "username": "string", "password_hash": "string", "email": "string", "role": "student"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>User not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div>

Code	Description
	<div>Example Value Schema</div> <div><pre>{ "message": "string" }</pre></div>

Users



GET /user/get Get users (public info)

Retrieves users with only public information. Authentication Basic



Parameters

Name	Description
username string (query)	<div>username</div>
role string (query)	<div>role</div>
email string (query)	<div>email</div>

Responses

Code	Description
200	<div>Array of users with public info</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{</pre></div>

Code	Description
	<pre>"_id": "string", "username": "string", "email": "string", "role": "student" }]</pre>
400	<div>Bad request</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "message": "string" }</div>



GET	/user/get/{id}	Get user by ID (public info)	 
Retrieves user with only public information. Authentication Basic			
Parameters			
Name	Description		

Name	Description
id ★ required string (path)	<div>id</div>



Responses

Code	Description
200	<div>User found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "username": "string", "email": "string", "role": "student"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>User not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div>



Code	Description
	<div>Example Value Schema</div> <pre>{ "message": "string" }</pre>

PUT	/user/update	Update own profile	 
Allows user to update their own email or password. Authentication Basic			
Parameters			
No parameters			
Request body		<div>application/json</div>	
<div>Example Value Schema</div> <pre>{ "email": "string", "password": "string" }</pre>			
Responses			
Code	Description		
200	Profile updated		
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div>		
	<div>Example Value Schema</div> <pre>{ "_id": "string", "username": "string", "email": "string", "role": "student" }</pre>		

Code	Description
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

DELETE	/user/delete	Delete own account	 
Permanently deletes the authenticated user's account. Authentication Basic			
Parameters			
No parameters			
Responses			
Code	Description		

Code	Description
200	Account deleted
403	Unauthorized
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
404	Not found
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

GET	/user/me	Get own profile	 
Retrieves the authenticated user's complete profile. Authentication Basic			
Parameters			
No parameters			
Responses			

Code	Description
200	<div>User profile</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "_id": "string", "username": "string", "email": "string", "role": "student"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string"}</pre></div>
404	<div>Not found</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string"}</pre></div>
500	<div>Server error</div> <div>Media type</div> <div>application/json</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "message": "string"}</pre></div>

Retrieves user by username with public information. Authentication Basic

Parameters

Name	Description
username * required string (path)	<div>username</div>

Responses

Code	Description
200	<div>User found</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "_id": "string", "username": "string", "email": "string", "role": "student"}</pre></div>
403	<div>Unauthorized</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "message": "string"}</pre></div>
404	<div>User not found</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{</pre></div>

Code	Description
	<pre> "message": "string" }</pre>
500	Server error
	Media type
	<div>application/json</div>
	Example Value Schema
	<pre>{ "message": "string" }</pre>

Schemas ^

Error {

message

string

Error message describing the issue

}

Assignment {

_id

string

Unique identifier for the assignment

title

string

Title of the assignment

description

string

Detailed description of the assignment

due_date

string(\$date-time)

Due date of the assignment

parent_group

string

ID of the group this assignment belongs to

}

```

CFAccount  {
    _id
        string
        CFAccount ID

    student_id
        string
        Associated student ID

    cf_account
        string
        CodeForces username

    is_verified_flag
        boolean
        Whether account has been verified
}

```

```

Challenge  {
    _id
        string
        Unique identifier for the challenge

    cf_code
        string
        CodeForces problem code

    student_id
        string
        ID of the student who attempted the challenge

    is_completed_flag
        boolean
        Indicates if the challenge has been completed

    completion_type
        string
        Type of completion ("contest" or "normal")
        Enum:
        [ contest, normal ]
}

```

```

DirectMessage  {
    _id
        string
        Unique identifier for the direct message

    sender_id
        string
        ID of the user who sent the message

    receiver_id
        string
        ID of the user who received the message

    message
        string
        Content of the direct message

    createdAt
        string($date-time)
        Timestamp when the message was created
}

```

```
Exercise    {
  _id      string
           Unique identifier for the exercise

  name     string
           Name of the exercise

  cf_code  string
           CodeForces problem code associated with the exercise

  parent_assignment string
           ID of the assignment this exercise belongs to
}
```

```
Following   {
  _id      string
           Unique identifier for the following relationship

  student_1_id string
           ID of the student who is following

  student_2_id string
           ID of the student being followed
}
```

```

Group    {
  _id      string
           Unique identifier for the group

  name     string
           Name of the group

  description string
           Description of the group

  parent_coach string
           ID of the coach who created the group

  group_messages [ {
                  sender_id string
                           ID of the user who sent the message

                  message string
                           Content of the message

                  timestamp string($date-time)
                           Timestamp when the message was sent
                } ]

  invite_code string
           Invite code for joining the group
}

```

```

StudentExercise {
  _id      string
           Unique identifier for the student exercise record

  student_id string
           ID of the student

  exercise_id string
           ID of the exercise

  completion_type string
           Type of completion ("contest" or "normal")
           Enum:
           [ contest, normal ]
}

```

```

StudentGroup  {
    _id
        string
        Unique identifier for the student-group relationship

    student_id
        string
        ID of the student

    group_id
        string
        ID of the group
}

```

```

User  {
    _id
        string
        Unique identifier for the user

    username
        string
        Username of the user

    password_hash
        string
        Hashed password of the user

    email
        string
        Email address of the user

    role
        string
        Role of the user in the system
        Enum:
        [ student, coach, admin ]
}

```

```

SafeUser  {
    _id
        string
        Unique identifier for the user

    username
        string
        Username of the user

    email
        string
        Email address of the user

    role
        string
        Role of the user in the system
        Enum:
        [ student, coach, admin ]
}

```

```
GroupMessage {  
  sender_id      string  
                 ID of the user who sent the message  
  
  message        string  
                 Content of the message  
  
  timestamp      string($date-time)  
                 Timestamp when the message was sent  
}
```