API Documentation

**URL for all calls: http://127.0.0.1:8000/**

Contents

[API calls for User 3](#_Toc166021715)

[Login 3](#_Toc166021716)

[Logout 3](#_Toc166021717)

[Create User 4](#_Toc166021718)

[Get User 5](#_Toc166021719)

[Delete User 6](#_Toc166021720)

[Update User 6](#_Toc166021721)

[API calls for Athletes 7](#_Toc166021722)

[Create Athlete 7](#_Toc166021723)

[Get Athlete 8](#_Toc166021724)

[Delete Athlete 8](#_Toc166021725)

[Update Athlete 9](#_Toc166021726)

[API Calls for Coaches 10](#_Toc166021727)

[Create Coach 10](#_Toc166021728)

[Get Coach 10](#_Toc166021729)

[Delete Coach 11](#_Toc166021730)

[API Calls for Teams 12](#_Toc166021731)

[Create Team 12](#_Toc166021732)

[Get Team 13](#_Toc166021733)

[Delete Team 14](#_Toc166021734)

[API Calls for Roster Management 15](#_Toc166021735)

[Add to Team 15](#_Toc166021736)

[View Roster 16](#_Toc166021737)

[Approve Athlete 17](#_Toc166021738)

[Remove Athlete 17](#_Toc166021739)

[API Calls for Workouts 18](#_Toc166021740)

[Create Workout 18](#_Toc166021741)

[Assign to Athlete(s) 19](#_Toc166021742)

[Get Workout 20](#_Toc166021743)

[Remove Athlete From Workout 21](#_Toc166021744)

[Edit Workout 21](#_Toc166021745)

[Copy Workout 22](#_Toc166021746)

[Delete Workout 22](#_Toc166021747)

[Get Athlete Workouts (Date Range) 23](#_Toc166021748)

[Get Coach Created Workouts (Date Range) 24](#_Toc166021749)

[API Calls for Training Groups 25](#_Toc166021750)

[Create Training Group 25](#_Toc166021751)

[Get Training Group 25](#_Toc166021752)

[Delete Training Group 26](#_Toc166021753)

[Add Athlete to Training Group 26](#_Toc166021754)

[Remove Athlete From Group 27](#_Toc166021755)

[API Calls for Activities 28](#_Toc166021756)

[Create Activity 28](#_Toc166021757)

[Get Athlete Activities 29](#_Toc166021758)

[Get Activity 30](#_Toc166021759)

[Delete Activity 31](#_Toc166021760)

[Edit Activity 32](#_Toc166021761)

[Get Activities Range 33](#_Toc166021762)

[API Calls for Stats 34](#_Toc166021763)

[Get Athlete Stats 34](#_Toc166021764)

[API Calls for Comments 35](#_Toc166021765)

[Create Comment 35](#_Toc166021766)

[Delete Comment 35](#_Toc166021767)

[Get Activity Comments 36](#_Toc166021768)

[API Calls For Strava 37](#_Toc166021769)

[Get Access Token 37](#_Toc166021770)

[Exchange Token 37](#_Toc166021771)

[Import Activities 38](#_Toc166021772)

[Revoke Access 39](#_Toc166021773)

# API calls for User

## Login

Extension: external/login

Request Type: POST

Parameters

* “username” (string)
* “password” (string)

Sample Request

/external/login?username=testAthlete&password=test

Successful Responses

{

"message": "Login successful",

"userId": 12,

"athleteId": 8

}

## Logout

Extension: external/logout

Request Type: POST

Parameters:

none

Sample Request

external/logout

Successful Response

{

"message": "Logout successful"

}

## Create User

Extension: external/newUser

Request Type: POST

Parameters:

“username” (string)

“password” (string)

“email” (string)

“isCoach” (Boolean)

“isAthlete (Boolean)

“first\_name” (string)

“last\_name” (string)

Sample Request

external/newUser?username=testAthlete3&password=test&email=testAthlete2%40email.com&isCoach=False&isAthlete=True&first\_name=test&last\_name=athlete2

Successful Response

{

"message": "User created successfully"

"userId": 12

}

## Get User

Extension: [external/getUser/<int:userID>/](http://127.0.0.1:8000/external/getUser/%3cint:userID%3e/)

Request Type: GET

Parameters

* None

Sample Request

/external/getUser/6/

Successful Response

{

"userData": {

"id": 13,

"username": "wyattw99",

"first\_name": "Wyatt",

"last\_name": "Wolfersberger",

"email": "wyattw99@gmail.com",

"isCoach": false,

"isAthlete": true

},

"athleteId": 9,

"teamId": null

}

## Delete User

Extension: [external/deleteUser/<int:userID>/](http://127.0.0.1:8000/external/deleteUser/%3cint:userID%3e/)

Request Type: DELETE

Parameters:

None

Successful Response

{

'message': 'User deleted successfully'

}

## Update User

Extension: external/updateUser/<int:userID/

Request Type: PUT or POST

Parameters:

“first\_name”

“last\_name”

“email”

Successful Response

{

“message”:”'User updated successfully”

"userId": 12

}

# API calls for Athletes

## Create Athlete

Extension: external/newAthlete

Request Type: POST

Parameters:

“birthday” (date, yyyy-mm-dd)

“schoolyear” (string)

“userID” (int)

Sample Request

/external/newAthlete?birthday=1999-10-05&schoolYear=Senior&userID=11

Successful Response

{

"message": "Athlete created successfully"

"athleteId": 8

}

## Get Athlete

Extension: external/getAthlete/<int:athleteID>/

Request Type: GET

Parameters:

None

Sample Request:

/external/getAthlete/4/

Successful Response

{

"id": 4,

"birthday": "1999-10-06",

"schoolYear": "Senior",

"teamID": 3,

"userID": 6

}

## Delete Athlete

Extension: external/deleteAthlete/<int:athleteID>/

Request Type: DELETE

Parameters:

None

Sample Request

/external/deleteAthlete/6/

Successful Response

{

"message": "Athlete deleted successfully"

}

## Update Athlete

Extension: external/updateAthlete/<int:athleteID>/

Request Type: PUT or POST

Parameters:

“birthday” (date, yyyy-mm-dd)

“schoolyear” (string)

Sample Request

/external/updateAthlete/7/?birthday=1999-10-05&schoolYear=Senior

Successful Response

{

"message": "Athlete updated successfully"

"athleteId": 8

}

# API Calls for Coaches

## Create Coach

Extension: external/newCoach

Request Type: POST

Parameters:

“userID” (int)

“teamID” (int)

Sample Request

external/newCoach?userId=9&teamId=3

Successful Response

{

"message": "Coach created successfully"

"coachId": 4

}

## Get Coach

Extension: external/getCoach/<int:coachID>/

Request Type: GET

Parameters:

None

Sample Request

/external/getCoach/4/

Successful Response

{

"id": 4,

"teamID": 3,

"userID": 8

}

## Delete Coach

Extension: external/deleteCoach/<int:coachID>

Request Type: DELETE

Parameters:

None

Sample Request

/external/deleteCoach/5/

Successful Response

{

"message": "Coach deleted successfully"

}

# API Calls for Teams

## Create Team

Extension: external/newTeam

Request Type: POST

Parameters:

“teamName” (string)

Sample Request

/external/newTeam?teamName=test+team3

Successful Response

{

"message": "Team created successfully"

"teamId": 4

}

## Get Team

Extension: external/getTeam/<str:teamName>/

Request Type: GET

Parameters:

None

Sample Request

external/getTeam/test team

Successful Response

{

"teamId": 3,

"teamName": "test team",

"coaches": [

{

"firstName": "test",

"lastName": "coach"

},

{

"firstName": "test",

"lastName": "coach2"

}

]

}

## Delete Team

Extension: external/deletedTeam/<int:teamID>/

Request Type: DELETE

Parameters:

none

Sample Request

external/deleteTeam/4/

Successful Response

{

"message": "Team deleted successfully"

}

# API Calls for Roster Management

## Add to Team

Extension: external/addToTeam/<int:athleteID>/

Request Type: PUT

Parameters:

“teamID” (int)

Sample Request

external/addToTeam/5/?teamId=3

Successful Response

{

"message": "Athlete added to team successfully"

}

## View Roster

Extension: external/viewRoster/<int:teamID>/

Request Type: GET

Parameters:

None

Sample Request

external/viewRoster/3/

Successful Response

{

"approved": [],

"pending": [

{

"firstName": "test",

"lastName": "Athlete",

"birthday": "1999-10-06",

"schoolyear": "Senior"

“athleteId”: 13

},

{

"firstName": "test",

"lastName": "athlete2",

"birthday": "1999-10-05",

"schoolyear": "Senior"

“athleteId”: 14

}

]

}

## Approve Athlete

Extension: external/approveAthlete/<int:athleteID>/

Request Type: PUT

Parameters:

None

Sample Request

external/approveAthlete/5/

Successful Response

{

"message": "Athlete approved successfully"

}

## Remove Athlete

Extension: external/removeFromTeam/<int:athleteID/

Request Type: PUT

Parameters:

None

Sample Request

external/removeFromTeam/5/

Successful Response

{

"message": "Athlete removed from team successfully"

}

# API Calls for Workouts

## Create Workout

Extension: external/newWorkout

Request Type: POST

Parameters:

“coachID” (int)

“description” (text)

“date” (date, yyyy-mm-dd)

“title” (char)

Sample Request

/external/newWorkout?coachID=4&description=test+to+see+if+workout+is+created&date=2024-04-14&title=test+workout

Successful Response

{

"message": "Workout created successfully",

"workoutID": 1

}

## Assign to Athlete(s)

Extension: external/assignToAthletes

Request Type: PUT

Parameters:

“workoutID” (int)

“numAthletes” (int)

“athleteIDs” (int,int,int,…)

Sample Request

/external/assignToAthletes?workoutID=1&numAthletes=2&athleteIDs=4%2C5%2C7

Successful Response

{

"message": "Athletes Assigned to Workout"

}

## Get Workout

Extension: external/getWorkout/<int:workoutID>/

Request Type: GET

Parameters:

none

Sample Request

/external/getWorkout/2/

Successful Response

{

"workoutID": 2,

"title": "test workout 2",

"description": "empty workout",

"assignedDate": "2024-04-14",

"coachID": 4,

"assignedAthletes": [

{

"athleteID": 4,

"firstName": "test",

"lastName": "Athlete"

}

]

}

## Remove Athlete From Workout

Extension: external/removeAthlete/<int:workoutID>/

Request Type: PUT

Parameters:

“athleteID” (int)

Sample Request

/external/removeAthlete/2/?athleteID=7

Successful Response

{

"message": "Athlete Removed from Workout"

}

## Edit Workout

Extension: external/editWorkout/<int:workoutID>/

Request Type: PUT

Parameters:

“title” (char)

“assignedDate” (date, yyyy-mm-dd)

“description” (text)

Sample Request

/external/editWorkout/2/?title=Edited+Workout&assignedDate=2024-04-14&description=This+workout+has+been+edited

Successful Response

{

"message": "Workout updated successfully",

"workoutID": 2

}

## Copy Workout

Extension: external/copyWorkout/<int:workoutID>/

Request Type: POST

Parameters:

“newDate” (date, yyyy-mm-dd)

“newTitle” (char)

Sample Request

/external/copyWorkout/2/?newDate=2024-10-05&newTitle=Copied+Workout

Successful Response

{

"message": "Workout copied successfully",

"workoutID": 3

}

## Delete Workout

Extension: external/deleteWorkout/<int:workoutID>/

Request Type: DELETE

Parameters:

none

Sample Request

/external/deleteWorkout/3/

Successful Response

{

"message": "Workout deleted successfully"

}

## Get Athlete Workouts (Date Range)

Extension: external/getWorkoutsRange/<int:athleteID>/

Request Type: GET

Parameters:

“rangeStart” (yyyy-mm-dd)

“rangeEnd” (yyyy-mm-dd)

Sample Request

/external/getWorkoutsRange/9/?rangeStart=2024-04-14&rangeEnd=2024-10-05

Successful Response

{

"assignedWorkouts": [

{

"workoutID": 5,

"title": "test workout ",

"description": "empty workout",

"assignedDate": "2024-04-30",

"coachID": 7

}

]

}

## Get Coach Created Workouts (Date Range)

Extension: external/getCoachWorkoutsRange/<int:coachID>/

Request Type: GET

Parameters:

“rangeStart” (yyyy-mm-dd)

“rangeEnd” (yyyy-mm-dd)

Sample Request

/external/getCoachWorkoutsRange/7/?rangeStart=2024-04-10&rangeEnd=2024-10-05

Successful Response

{

"assignedWorkouts": [

{

"workoutID": 5,

"title": "test workout ",

"description": "empty workout",

"assignedDate": "2024-04-30",

"coachID": 7,

"assignedAthletes": [

{

"athleteID": 9,

"firstName": "Wyatt",

"lastName": "Wolfersberger"

},

{

"athleteID": 10,

"firstName": "test",

"lastName": "user"

}

]

}

]

}

# API Calls for Training Groups

## Create Training Group

Extension: external/newTrainingGroup

Request Type: POST

Parameters:

“groupName” (string)

“teamID” (int)

Sample Request

/external/newTrainingGroup?groupName=Wyatts+Group&teamID=5

Successful Response

{

"message": "Group created successfully",

"groupId": 1

}

## Get Training Group

Extension: external/getTrainingGroup/<int:groupID>/

Request Type: GET

Parameters:

none

Sample Request

/external/getTrainingGroup/1/

Successful Response

{

"groupId": 1,

"groupName": "Wyatts Group",

"athletes": []

}

## Delete Training Group

Extension: external/deleteTrainingGroup/<int:groupID>/

Request Type: DELETE

Parameters:

none

Sample Request

/external/deleteTrainingGroup/1/

Successful Response

{

"message": "Group deleted successfully"

}

## Add Athlete to Training Group

Extension: external/addAthleteToGroup/<int:athleteID>/

Request Type: PUT

Parameters:

“groupID” (int)

Sample Request

/external/addAthleteToGroup/9/?groupID=1

Successful Response

{

"message": "Athlete added to group successfully"

}

## Remove Athlete From Group

Extension: external/removeAthleteFromGroup/<int:athleteID>

Request Type: PUT

Parameters:

“groupID” (int)

Sample Request

/external/removeAthleteFromGroup/9/?groupID=1

Successful Response

{

"message": "Athlete removed from team successfully"

}

# API Calls for Activities

## Create Activity

Extension: external/newActivity

Request Type: POST

Parameters:

“athleteID” (int)

“type” (string)

“description” (text)

“name” (string)

“movingTime” (int)

“elapsedTime” (int)

“distance” (float)

“hasHeartrate” (Boolean)

“manual” (Boolean)

“maxPace” (float)

“averageCadence” (float)

“startDate” (datetime)

Sample Request

/external/newActivity?athleteID=9&type=run&description=this+a+test+activity&name=test+run+&movingTime=300&elapsedTime=330&distance=1609&hasHeartrate=False&manual=True&maxPace=10.0&averageCadence=160.0&startDate=2024-04-14+17%3A38%3A00

Successful Response

{

"message": "Run created successfully",

"activityId": 38

}

## Get Athlete Activities

Extension: external/getAthleteActivities/<int:athleteID>/

Request Type: GET

Parameters:

“activityType” (string)

Sample Request

/external/getAthleteActivities/9/?activityType=all

Successful Response

{

"bikeData": [],

"runData": [

"Id: 38, Activity: test run , Type: run, Time: 300.0, Start Date: 2024-04-14 17:38:00+00:00, Distance: 1609.0"

],

"swimData": [],

"otherData": []

}

## Get Activity

Extension: external/getActivity/<int:activityID>/

Request Type: GET

Parameters:

“activityType” (string)

Sample Request

/external/getActivity/38/?activityType=run

Successful Response

{

"athlete": "9",

"activityType": "run",

"name": "test run ",

"description": "this a test activity",

"movingTime": "300.0",

"elapsedTime": "330.0",

"startDate": "2024-04-14 17:38:00+00:00",

"distance": "1609.0",

"averageHeartrate": "0",

"maxHeartrate": "0",

"averagePace": "5.363333333333333",

"maxSpeed": "10.0",

"averageCadence": "160.0"

}

## Delete Activity

Extension: external/deleteActivity/<int:activityID>/

Request Type: DELETE

Parameters:

“activityType” (string)

Sample Request

/external/deleteActivity/1/?activityType=swim

Successful Response

{

"message": "Activity deleted successfully"

}

## Edit Activity

Extension: external/editActivity/<int:activityID>/

Request Type: PUT

Parameters:

“activityType” (string)

“description” (text)

“name” (string)

“movingTime” (int)

“elapsedTime” (int)

“distance” (float)

“hasHeartrate” (Boolean)

“manual” (Boolean)

“maxPace” (float)

“averageCadence” (float)

“startDate” (datetime)

Sample Request

/external/editActivity/1/?activityType=bike&description=this+a+test+activity&name=test+bike+upadated&movingTime=310&elapsedTime=330&distance=1609&hasHeartrate=False&manual=True&maxPace=10.0&averageCadence=160.0&startDate=2024-04-15+17%3A38%3A00&maxSpeed=10.0

Successful Response

{

"message": "Activity updated successfully",

"activityId": 1

}

## Get Activities Range

Extension: external/getAthleteActivitiesRange/<int:athleteID>/

Request Type: GET

Parameters:

“rangeStart” (yyyy-mm-dd)

“rangeEnd” (yyyy-mm-dd)

“activityType” (string)

Sample Request

/external/getAthleteActivitiesRange/9/?rangeStart=2024-04-13&rangeEnd=2024-04-15&activityType=all

Successful Response

{

"bikeData": [

"Id: 1, Activity: test bike upadated, Type: bike, Time: 310.0, Start Date: 2024-04-15 17:38:00+00:00, Distance: 1609.0"

],

"runData": [

"Id: 38, Activity: test run , Type: run, Time: 300.0, Start Date: 2024-04-14 17:38:00+00:00, Distance: 1609.0"

],

"swimData": [],

"otherData": []

}

# API Calls for Stats

## Get Athlete Stats

Extension: external/getAthleteStats/<int:athleteID>/

Request Type: GET

Parameters:

“activityType” (string)

“rangeStart” (yyyy-mm-dd)

“rangeEnd” (yyyy-mm-dd)

Sample Request

/external/getAthleteStats/9/?activityType=all&rangeStart=2024-04-10&rangeEnd=2024-04-17

Successful Response

{

"bikeData": {

"numBikes": 1,

"Distance": 1609,

"Time": 310,

"averageSpeed": 5.190322580645161,

"averageDistance": 1609,

"averageHeartrate": 0

},

"runData": {

"numRuns": 1,

"Distance": 1609,

"Time": 300,

"averagePace": 5.363333333333333,

"averageDistance": 1609,

"averageHeartrate": 0

},

"swimData": {},

"otherData": {}

}

# API Calls for Comments

## Create Comment

Extension: external/createComment

Request Type: POST

Parameters:

“activityType” (string)

“activityID” (int)

“text” (text)

Sample Request

/external/createComment?activityType=run&activityID=38&text=Great+job

Successful Response

{

"message": "Comment created successfully",

"commentId": 1

}

## Delete Comment

Extension: external/deleteComment/<int:commentID>/

Request Type: DELETE

Parameters:

none

Sample Request

/external/deleteComment/2/

Successful Response

{

"message": "Comment deleted successfully"

}

## Get Activity Comments

Extension: external/getComments/<int:activityID/"

Request Type: GET

Parameters:

“activityType” (string)

Sample Request

/external/getComments/38/?activityType=run

Successful Response

{

"comments": [

{

"commentId": 1,

"text": "Great job"

},

{

"commentId": 2,

"text": "Do better"

}

]

}

# API Calls For Strava

## Get Access Token

Extension: external/getStravaAccessToken

Request Type: GET

Parameters:

“athleteID” (int)

Sample Request

/external/getStravaAccessToken?athleteID=9

Successful Response

{

"redirectURL": "http://www.strava.com/oauth/authorize?client\_id=98300&response\_type=code&redirect\_uri=http://127.0.0.1:8000/external/9/exchangeToken&approval\_prompt=force&scope=activity:read\_all"

}

## Exchange Token

Extension: external/<int:athleteID>/exchangeToken

Request Type: POST

Parameters:

none

Sample Request

/external/9/exchangeToken?state=&code=9a37442754b0cd913aae08110a28ba7f8b6206a9&scope=read,activity:read\_all

Successful Response

{

"Status": "linked",

"message": "Account Linked, please close this page"

}

## Import Activities

Extension: external/getStravaActivities

Request Type: GET

Parameters:

“athleteID” (int)

“rangeStart” (yyyy-mm-dd)

“rangeEnd” (yyyy-mm-dd)

Sample Request

/external/getStravaActivities?athleteID=9&rangeStart=2024-05-01&rangeEnd=2024-05-08

Successful Response

{

"message": "Strava activities added to database",

"databaseActivities": [

{

"name": "Afternoon Run",

"activityId": 39

},

{

"name": "Lunch Run",

"activityId": 40

},

{

"name": "Morning Run",

"activityId": 41

},

{

]

}

## Revoke Access

Extension: external/revokeStravaAccess

Request Type: DELETE

Parameters:

“athleteID” (int)

Sample Request

/external/revokeStravaAccess?athleteID=9

Successful Response

{

"access\_token": "05439d682b4779e0e4718527f795027bbaf10eb8"

}