

Testing takes place at the URL: <http://shadowrealm.cse.mssate.edu/gsas/API/>

In test URIs, something with a colon (:) in front of it is a variable and can change with every request.

All URIs present will be provided. A returned 400 for bad request specifies that improper JSON was sent to the server.

COURSES

Resource	Description	Data Format
GET /courses	<p>Lists all the available courses.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{“courses”: [{“course”: { “id”: integer, “name”: string, “location”: string } }, {“course”: { “id”: integer, “name”: string, “location”: string } }]}</pre>
GET /courses/:id	<p>List the information on a specific course.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error 204 - No Content the course doesn’t exist</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{“course”: { “id”: integer, “name”: string, “location”: string }</pre>

Resource	Description	Data Format
POST /courses	<p>Insert a new course into the database.</p> <p>Return code: 201 - Created</p> <p>Errors: 5xx - Server errors 400 - Bad Request</p>	<p>SEND DATA:</p> <p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "course": { "id": null, "name": string, "location": string } }</pre> <p>RETURN DATA:</p> <p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "course": { "id": integer, "name": string, "location": string } }</pre>
POST /courses/:id	<p>Update a course specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server errors 400 - Bad Request 204 - No content trying to update a course that doesn't exist</p>	<p>Send and return data is the same.</p> <p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "course": { "id": integer, "name": string, "location": string } }</pre>

Resource	Description	Data Format
POST /courses/destroy/:id	<p>Deletes a course from the database specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server errors 204 - No content trying to delete a course that doesn't exist</p>	<p>Send and return data is the same.</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "course": { "id": integer, "name": string, "location": string } }</pre>

USERS

Resource	Description	Data Format
GET /users	<p>List all the users in the database.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "users": [{ "user": { "id": integer, "name": string, }, "user": {} }] }</pre>
GET /users/:id	<p>List the information of a specific user as specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error 204 - No Content the User doesn't exist</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "users": [{ "user": { "id": integer, "memberID": string, "nickname": string, "name": string, "email": string, "birthdate": string, "gender": string, "rightHanded": boolean, "stats": { "year": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } } }, "user": {} }] }</pre>

Resource	Description	Data Format
POST /users	<p>Insert a new User into the database.</p> <p>Return code: 201 - Created</p> <p>Errors: 5xx - Server error 400 - Bad Request 412 - Email and member ID already taken 406 - Member ID already taken 409 - Email already taken</p>	<p>SEND DATA:</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "user": { "memberID": string, "nickname": string, "name": string, "email": string } }</pre> <p>RETURN DATA:</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "users": [{ "user": { "id": integer, "memberID": string, "nickname": string, "name": string, "email": string, "birthdate": string, "gender": string, "rightHanded": boolean, "stats": { "year": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } } } }, { "user": {} }] }</pre>

Resource	Description	Data Format
POST /users/:id	<p>Update an existing User specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors:</p> <ul style="list-style-type: none"> 5xx - Server error 400 - Bad Request 412 - Email and member ID already taken 406 - Member ID already taken 409 - Email already taken 204 - No content trying to update a User that doesn't exist 	<p>Send and return data is the same.</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "users": [{ "user": { "id": integer, "memberID": string, "nickname": string, "name": string, "email": string, "birthdate": string, "gender": string, "rightHanded": boolean, "stats": { "year": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } } } }], "user": {} }</pre>

Resource	Description	Data Format
POST /users/destroy/:id	<p>Deletes a User from the database specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server errors 204 - No content trying to delete a User that doesn't exist</p>	<p>Send and return data is the same.</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "users": [{ "user": { "id": integer, "memberID": string, "nickname": string, "name": string, "email": string, "birthdate": string, "gender": string, "rightHanded": boolean, "stats": { "year": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } } } }], "user": {} }</pre>

HOLES

Resource	Description	Data Form
GET /holes/ reference/:courseID/:teeID	<p>Get reference points for all holes in a specified course with a certain tee.</p> <p>Return code: 200 - OK</p> <p>Errors: 400 - Bad tee or course 5xx - Server error</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "holes": [{ "hole": { "distance": int, "par": int, "holeNumber": int, "firstRefLat": double, "firstRefLong": double, "secondRefLat": double, "secondRefLong": double, "thirdRefLat": double, "thirdRefLong": double, "firstRefX": int, "firstRefY": int, "secondRefX": int, "secondRefY": int, "thirdRefX": int, "thirdRefY": int } }, { "hole": {} }] }</pre>

ROUNDS

Resource	Description	Data Form
GET /all/:page	<p>List all rounds in the database. Supports pagination with page numbers.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "rounds": [{ "round": { "id": integer, "startTime": timestamp, }, { "round": {} }], "nextPage": int or null }</pre>
GET /rounds/:id	<p>List information of a hole specified by the provided ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error 204 - No content The round doesn't exist</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre>{ "round": { "id": integer, "course": { "id": integer, "name": string, "location": string }, "totalScore": integer, "teeID": integer, "startTime": timestamp, "holes": [] } }</pre>

Resource	Description	Data Form
GET /rounds/ user/:id/:page	<p>List all the rounds of a User. The user is specified by the ID.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server error 204 - Not found the User doesn't exist</p>	<p>[‘Content-Type’] = ‘appliaction/json’</p> <pre> {"rounds": [{ "round": { "id": integer, "startTime": timestamp, }, { "round": {} }], "nextPage": int or null } </pre>

Resource	Description	Data Form
POST /rounds	<p>Insert a new hole into the database.</p> <p>Return code: 201 - Created</p> <p>Errors: 400 - Bad Request 5xx - Server errors</p>	<p>SEND DATA:</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "round": { "user": { "id": integer, "username": string, "password": string, "name": string, "email": string, "roles": integer, "stats": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } }, "course": { "id": integer, "name": string, "location": string } }, "totalScore": integer, "teeID": integer, "startTime": timestamp, "holes": [] }</pre>

Resource	Description	Data Form
		<p>RETURN DATA:</p> <p>[‘Content-Type’] = ‘appliaction/ json’</p> <pre>{ "round": { "id": integer, "course": { "id": integer, "name": string, "location": string } }, "totalScore": integer, "teeID": integer, "startTime": timestamp, "holes": [] }</pre>

Resource	Description	Data Form
POST /rounds/:id	<p>Update a hole.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx - Server errors 400 - Bad Request 204 - No content trying to update a round that doesn't exist</p>	<p>Send and return data is the same.</p> <p>['Content-Type'] = 'appliaction/json'</p> <pre>{ "round": { "id": integer, "user": { "id": integer, "username": string, "password": string, "name": string, "email": string, "roles": integer, "stats": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } }, "course": { "id": integer, "name": string, "location": string } }, "totalScore": integer, "teeID": integer, "startTime": timestamp, "holes": [] }</pre>

Resource	Description	Data Form
POST /rounds/ destroy/:id	<p>Delete a hole from the database.</p> <p>Return code: 200 - OK</p> <p>Errors: 5xx -Server errors 204 - No content trying to delete a hole that doesn't exist</p>	<p>Send and return data is the same.</p> <p>['Content-Type'] = 'appliaction/ json'</p> <pre>{ "round": { "id": integer, "user": { "id": integer, "username": string, "password": string, "name": string, "email": string, "roles": integer, "stats": { "driving_distance": double, "GIR_percentage": double, "driving_accuracy": double } }, "course": { "id": integer, "name": string, "location": string } }, "totalScore": integer, "teeID": integer, "startTime": timestamp, "holes": [] }</pre>