*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Feature Document

User Story #**270**

**Team Member:**

Gabriel Fernandez

**Product Owner(s)**:

Mohsen Taheri

**Mentor(s)**:

Mohsen Taheri

**Instructor**: Masoud Sadjadi

**User Story** Create Object-level Permissions in the Backend

* As a front-end developer, I want to make sure all the data I get from the database pertains only to the logged in user, so that the data is encapsulated.

**Acceptance Criteria**

* Authentication must be turned on for the database
* The database must only return information relevant to the logged in user
* A user must not be able to edit or delete other users’ database models

**Use Case** #**001 – Fetch Logged-in User’s Data**

**Actors**

Front-end developer

**Entry Conditions**

Front-end developer has access to a terminal or an api to send http requests

Front-end developer has read the documentation and understands which api endpoints to hit and what data to send

**Flow of Events**

1. Use case starts when front-end developer sends an http request to any backend endpoint with a user token as a header and any information the endpoint may need
2. The backend authenticates the user
3. The backend saves the user key
4. The backend queries the database for the information asked from the respective model
5. The backend filters the results by the saved user key
6. The backend returns the filtered data to the front-end developer as a response, and the use case ends

**Alternate Flow of Events**

* 2a.
  + The backend determines the token does not belong to an existing user and the use case ends.
* 4a.
  + The backend determines the input is not correct, and returns a response with the error and the use case ends.

**Use Case Diagram**



**Sequence Diagram**



**Class Diagram**



**Unit Test**

**Test Case 1 (Sunny Day)**

**Purpose**

* Ensure backend returns only the data that belongs to the logged in user

**Precondition**

* Http request is made to the server at the dashboard endpoint

**Input**

* User token

**Expected Result**

* Array of assignments & tasks that belong to the user

**Actual Result**

* Array of assignments & tasks that belong to the user

**Test Case 2 (Rainy Day)**

**Purpose**

* Ensure that a user cannot delete an object that they did not create

**Precondition**

* Http request is made to the server at the class endpoint, with DELETE method

**Input**

* User token, Class PK

**Expected Result**

* Error message saying that the user doesn’t have permissions to perform that action

**Actual Result**

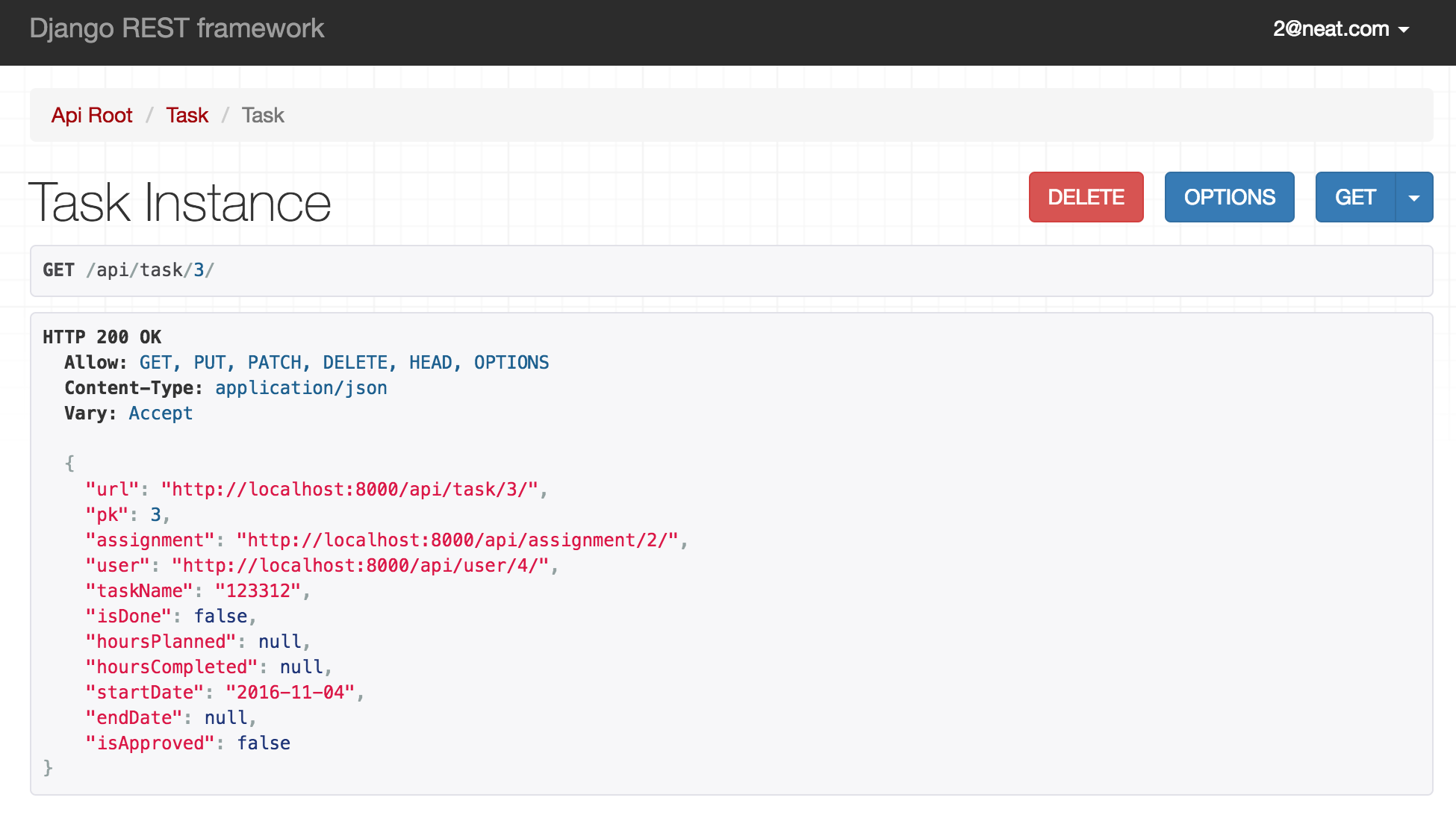
* Error message saying that the user doesn’t have permissions to perform that action

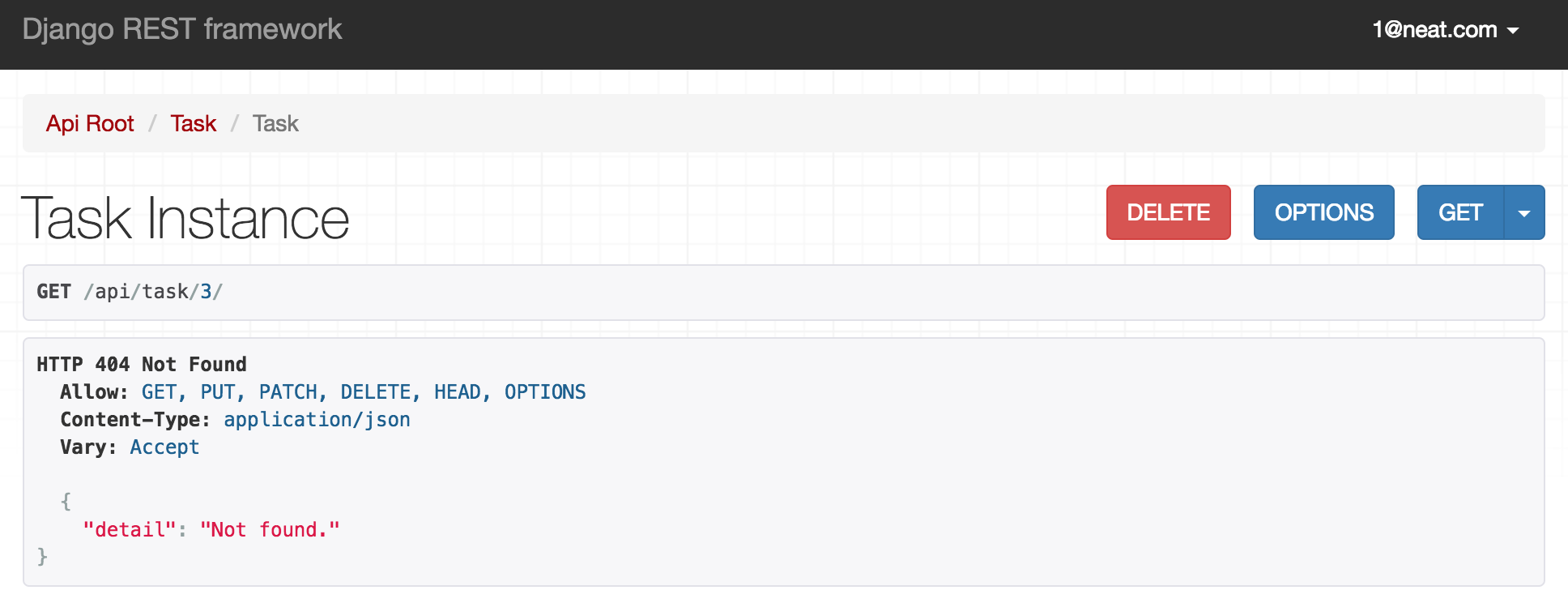
**Visual User Guide**

Getting only data that pertains to logged in user:

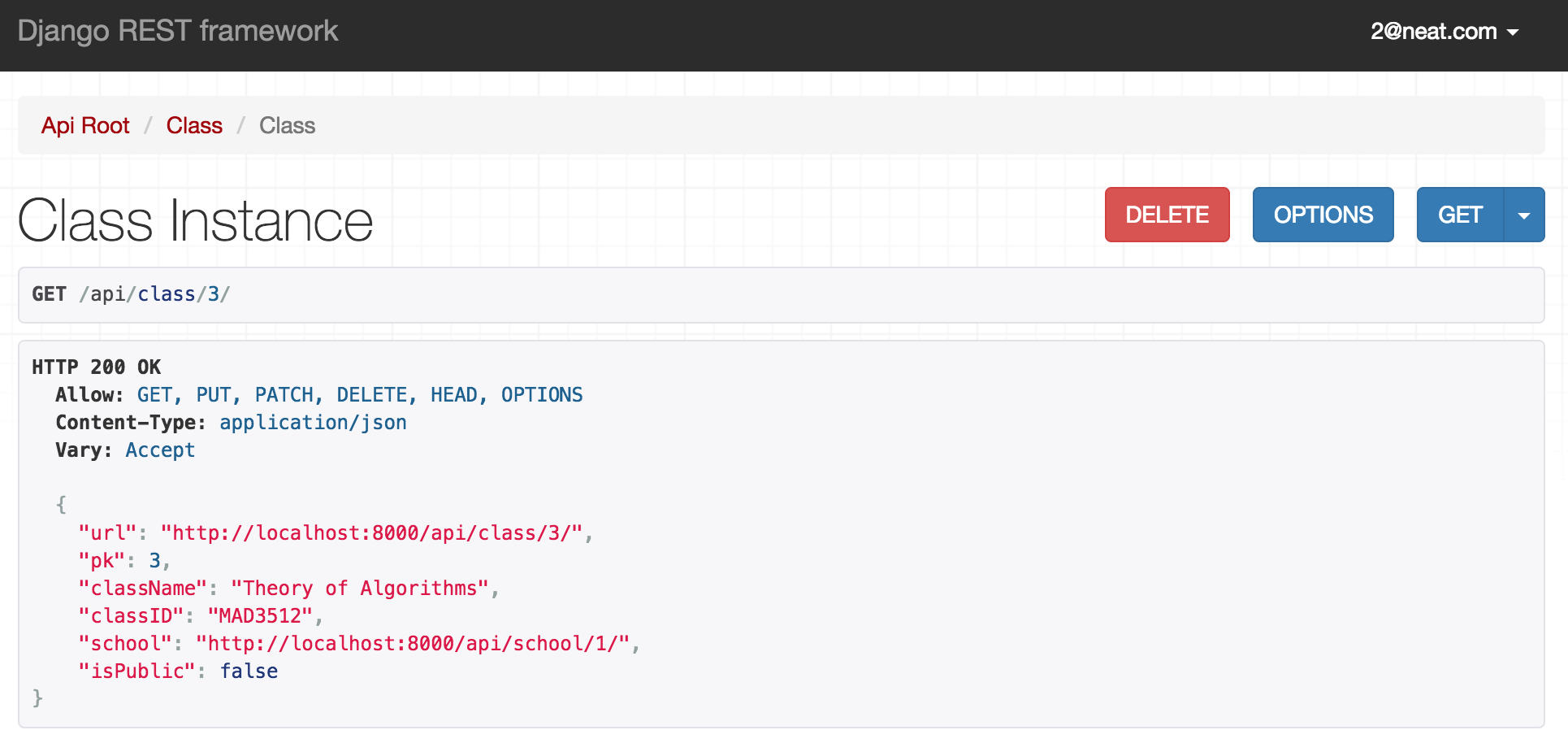


Not allowing other users to view your data; The same task appears as non-existent to another user:





You can delete or edit objects you created, but other users cannot:



* After attempting to delete the same class instance as another user:

