Design: Compliance Enhancement System for Santam

Context

This page outlines the design for Compliance Enhancement System for Santam

Scope

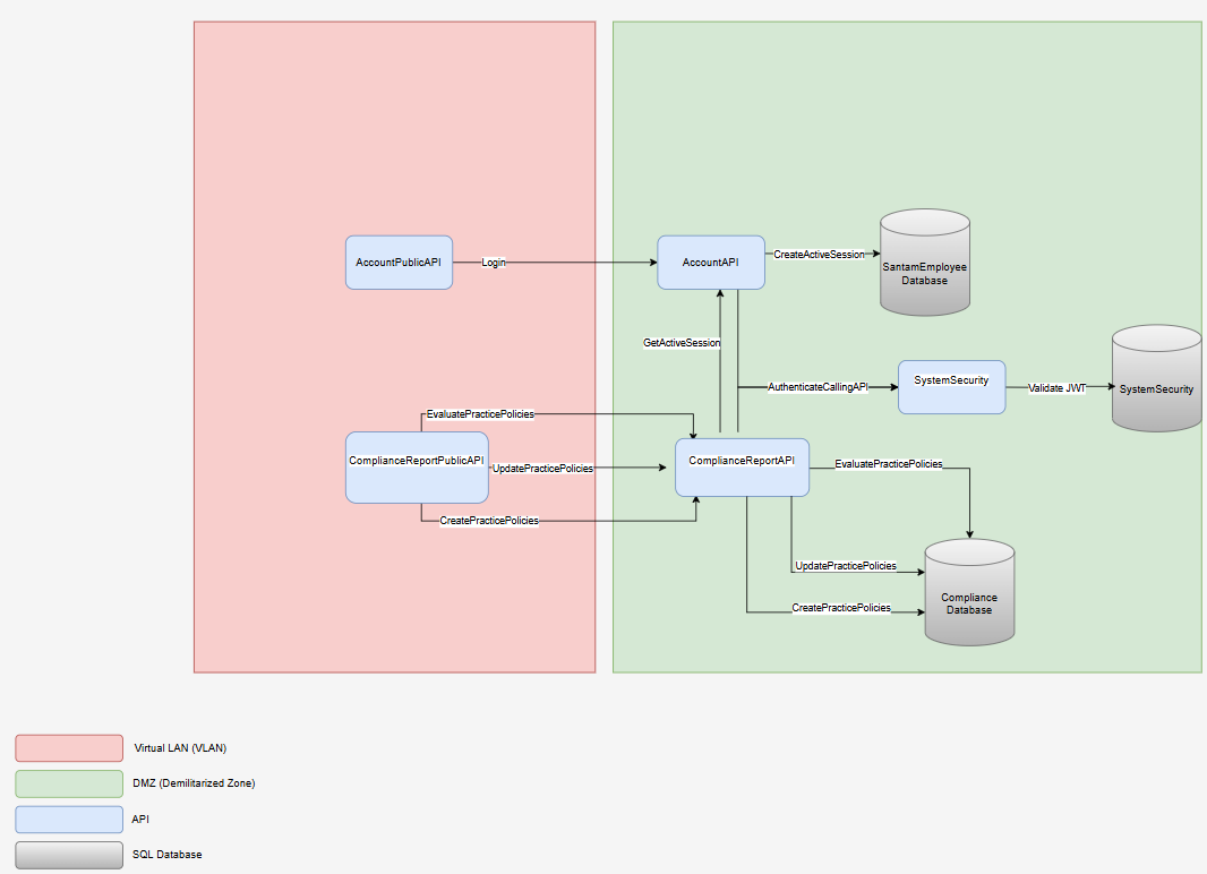
The scope of the Compliance Enhancement System includes:

- AccountPublicAPI
  - Login
- AccountAPI
  - Login
  - CreateActiveSession
  - GetActiveSession
- ComplianceReportPublicAPI
  - EvaluatePracticePolicy
  - UpdatePracticePolicy
  - CreatePracticePolicy
- ComplianceReportAPI
  - EvaluatePracticePolicy
  - UpdatePracticePolicy
  - CreatePracticePolicy
- Compliance database

Assumptions

- SystemSecurity system exists.
- Santam Employees database exists

High Level Design



The design above indicates that there are 5 API required for the Compliance Enhancement System to work properly with an assumption that Santam Employee database and SystemSecurityAPI are already in place.

The user of the Compliance Enhancement System is required to login before being able to Create, Update and Evaluate (retrieve) practice policies.

Login

- AccountPublicAPI is facing the public internet. This API is responsible for executing login requests from public internet client(s). The API endpoint validates the request structure if it meets the expectations of a login request.
    - **Login Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body

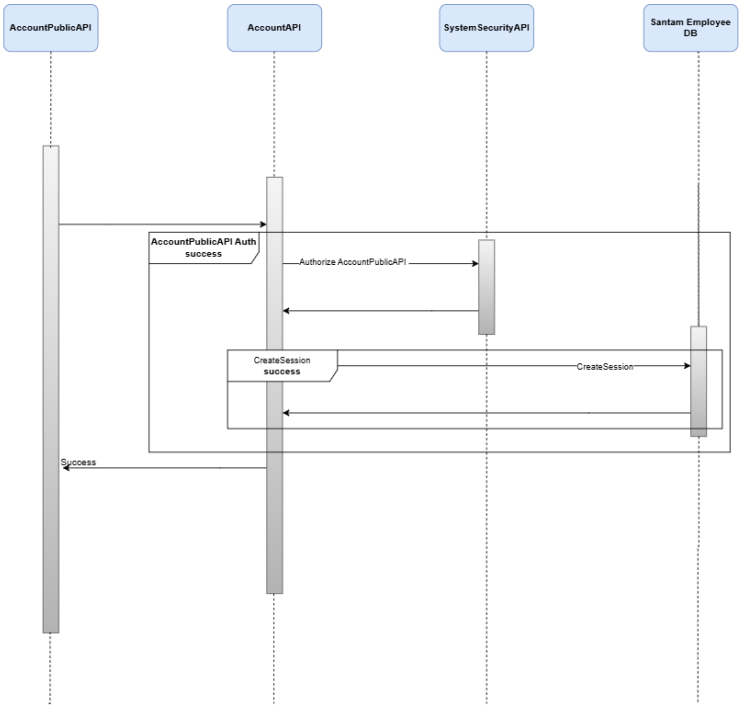
        - Username: username
        - Password: password
    - Response: Status code: 200, or 400 indicating success of bad request respectively

- AccountAPI is in the more secured network (DMZ). This API is responsible for authenticating the AccountPublicAPI and create the session for the account to be logged in. Session can only be created if the account credentials were successfully validated.
    - **SystemSecurityAPI Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - APIAuthToken: jwt
        - Response: Status code: 200, or 400 indicating success of bad request respectively

    - **CreateSession Request**
        - Using the connection string in configured in the appsettings.json, call the stored procedure to validate the account. For example,
            - EXECUTE dbo.pr_ValidateAccount @username = @username, @password = @password
            - Response would be a user ID
        - Using the connection string in configured in the appsettings.json, call the stored procedure to create session. For example,
            - EXECUTE dbo.pr_CreateSession @userId = @userId

Login Sequence Diagram

See the sequence diagram below for login



Evaluate Practice Policies

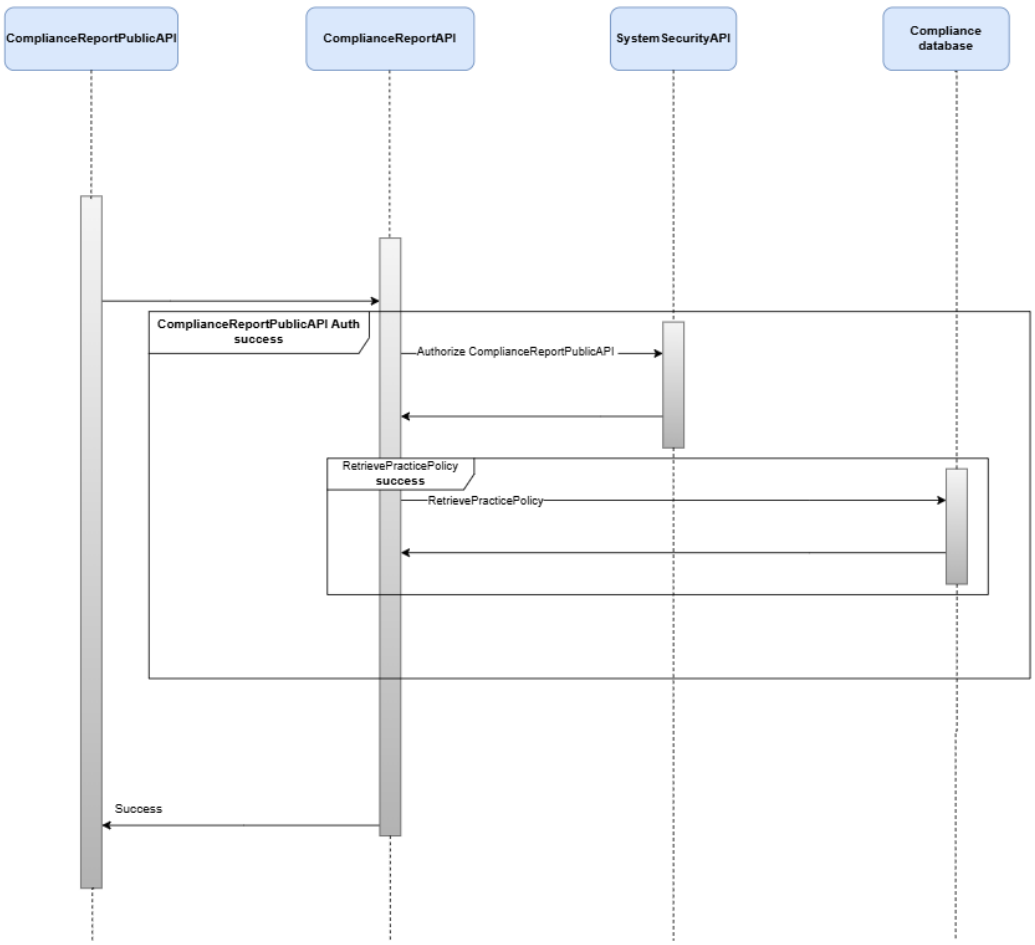
- CompliancePublicAPI is facing the public internet. This API is responsible for executing evaluation, creation and updating of practice polices requests from public internet client(s). This section focuses on EvaluatePracticePolicy endpoint. This endpoint validates the request structure if it meets the expectations of a EvaluatePracticePolicy request.
    - **EvaluatePracticePolicy Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - practiceId: practiceId
            - policyId: policyId
            - userId: userId
    - **EvaluatePracticePolicy Response**
        - List of policies for the practice
        - Response
            - Status code: 200, or 400 indicating success of bad request respectively
- ComplianceAPI is in the more secured network (DMZ). This API is responsible for authenticating the CompliancePublicAPI and retrieving practice policy as per request.
    - **SystemSecurityAPI Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - APIAuthToken: jwt
    - **AccountAPI GetActiveSession Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - UseId: UserId
        - Using the connection string in configured in the appsettings.json, call the stored procedure to GetSession session. For example,
            - EXECUTE dbo.pr_GetUserSession @userId = @userId
        - Response

        - Status code: 200, or 400 indicating success of bad request respectively
    - **Retrieve PracticePolicy Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - practiceId: practiceId
            - policyId: policyId

        - Using the connection string in configured in the appsettings.json, call the stored procedure to retrieve practice policy. For example,
            - EXECUTE dbo.pr_RetrievePracticePolicy @ practiceId = @ practiceId, @ policyId = @ policyId
        - Response
            - List of policies for the practice

EvaluatePracticePolices Sequence Diagram

See the sequence diagram below for Evaluate Practice Policy



Update Practice Policies

- CompliancePublicAPI is facing the public internet. This API is responsible for executing evaluation, creation and updating of practice policy requests from public internet client(s). This section focuses on UpdatePracticePolicy endpoint. This endpoint validates the request structure if it meets the expectations of a UpdatePracticePolicy request.
    - **UpdatePracticePolicy Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - practiceId: practiceId
            - policyId: policyId
            - policyUpdate: policyUpdate
            - userId: userId
        - Response

- Status code: 200, or 400 indicating success of bad request respectively

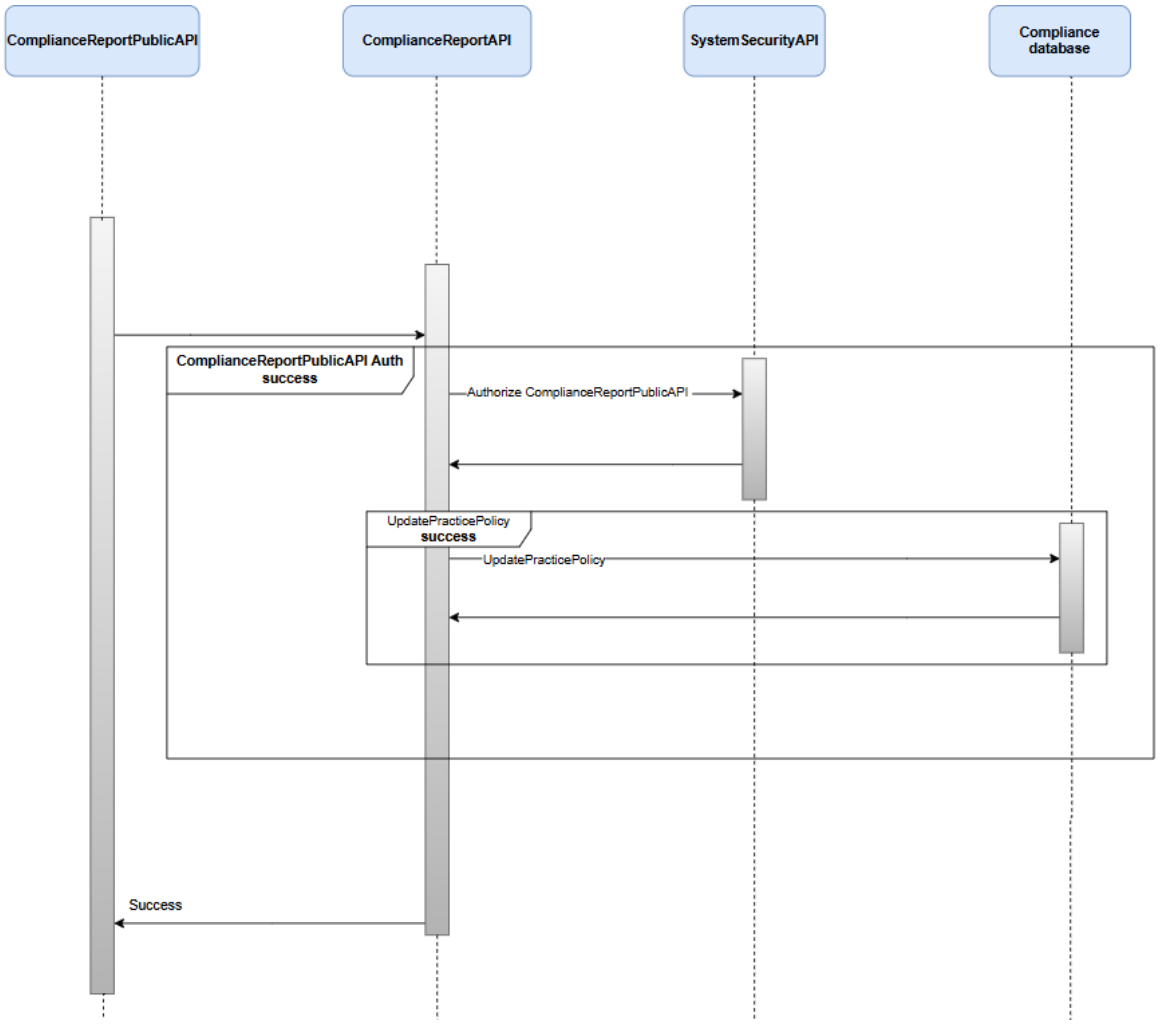
- ComplianceAPI is in the more secured network (DMZ). This API is responsible for authenticating the CompliancePublicAPI and update practice policy as per request.
    - **SystemSecurityAPI Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - APIAuthToken: jwt
    - **AccountAPI GetActiveSession Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - UseId: UserId
        - Using the connection string in configured in the appsettings.json, call the stored procedure to GetSession session. For example,
            - EXECUTE dbo.pr_GetUserSession @userId = @userId
        - Response
            - Status code: 200, or 400 indicating success of bad request respectively

    - **UpdatePracticePolicy Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - practiceId: practiceId
            - policyId: policyId
            - policyUpdate: policyUpdate
            - userId: userId
        - Response
            - Status code: 200, or 400 indicating success of bad request respectively

- Using the connection string in configured in the appsettings.json, call the stored procedure to update practice policy. For example,
    - EXECUTE dbo.pr_UpdatePracticePolicy @ practiceId = @ practiceId, @ policyId = @ policyId, @ policyUpdate = @ policyUpdate
- Response code = 1 indicating a success

UpdatePracticePolicy Sequence Diagram

See the sequence diagram below for Update Practice Policy



Create Practice Policies

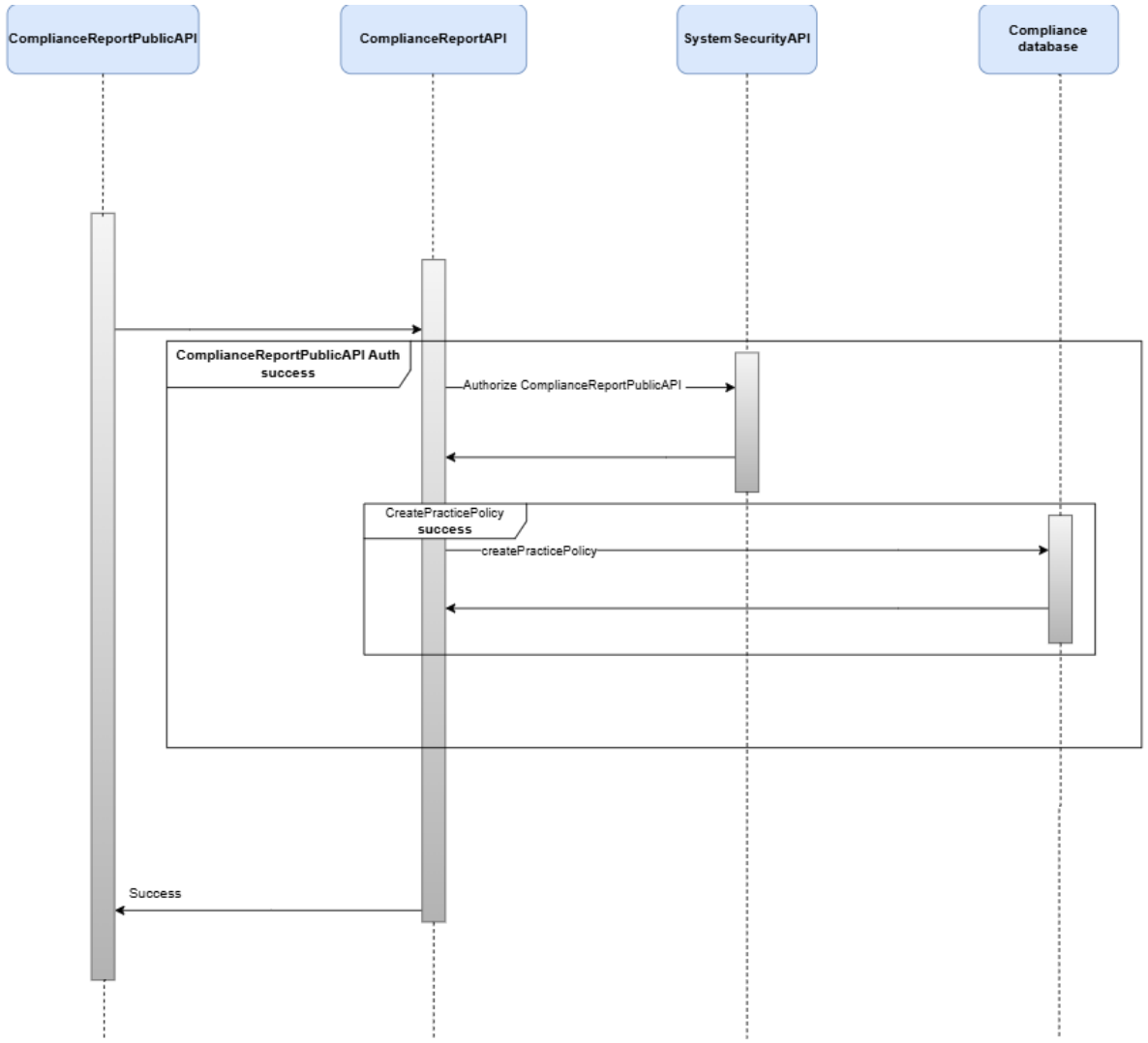
- CompliancePublicAPI is facing the public internet. This API is responsible for executing evaluation, creation and updating of practice polices requests from public internet client(s). This section focuses on CreatePracticePolicy endpoint. This endpoint validates the request structure if it meets the expectations of a CreatePracticePolicy request.
    - **CreatePracticePolicy Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - practiceId: practiceId
            - policyId: policyId
            - newPolicy: newPolicy
            - userId: userId
    - **CreatePracticePolicy Response**
        - Response: Status code: 200, or 400 indicating success of bad request respectively

- ComplianceAPI is in the more secured network (DMZ). This API is responsible for authenticating the CompliancePublicAPI and creating practice policy as per request.
    - **SystemSecurityAPI Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - APIAuthToken: jwt
    - **AccountAPI GetActiveSession Request**
        - Headers
            - ContentType: application/json
            - Authorization: Bearer JWT
        - Body
            - UseId: UserId
        - Using the connection string in configured in the appsettings.json, call the stored procedure to GetSession session. For example,
            - EXECUTE dbo.pr_GetUserSession @userId = @userId
        - Response: Status code: 200, or 400 indicating success of bad request respectively

- **CreatePracticePolicy Request**
  - Using the connection string in configured in the appsettings.json, call the stored procedure to create practice policy. For example,
    - EXECUTE dbo.pr_CreatePracticePolicy @ practiceId = @ practiceId, @ policyId = @ policyId, @newPolicy = @newPolicy

Response code = 1 indicating a success

CreatePracticePolicy Sequence Diagram



Testing

Unit tests

- AccountPublicAPI
    - Login
- AccountAPI
    - Login
    - GetSession
- ComplianceReportPublicAPI
    - EvaluatePracticePolicy
    - UpdatePracticePolicy
    - CreatePracticePolicy
- ComplianceReportAPI
    - EvaluatePracticePolicy
    - UpdatePracticePolicy
    - CreatePracticePolicy
- tSQlt
    - dbo.pr_GetUserSession
    - dbo.pr_UpdatePracticePolicy
    - dbo.pr_RetrievePracticePolicy
    - dbo.pr_CreatePracticePolicy
    - dbo.pr_ValidateAccount
    - dbo.pr_CreateSession

Integration testing

- Login flow
- CreatePracticePolicy flow
- UpdatePracticePolicy flow
- EvalutatePracticePolicy flow

Load performance

Deployment

Deployment to be done in the following sequence

1. All database patches
2. All API in the DMZ
3. All API in VLan

Monitoring

1. Error codes in the login endpoint
2. Error codes in the Evaluate practice policy endpoint
3. Error codes in the Update practice policy endpoint
4. Error codes in the Create practice policy endpoint