```
swagger: '2.0'
info:
  title: Authentication API
  description: Use the Authentication API to generate, refresh, and manage JSON Web Tokens (JWT) that are required for
authorization in all Enterprise Control Room APIs.


    The JWT authorization is based on the roles and permission of the user's account that is used in the request body.

        - An authentication token's default time out is 20 minutes.

        >> **Note:** The default timeout for tokens is configurable. For more information about authentication methods, see
[Supported authentication methods](https://docs.automationanywhere.com/csh?context=csh-supported-authentication-methods).

        - Include the JWT in a header file for requests in other Control Room APIs.

        - **Header file example:** `-H X-Authorization <JSON Web Token>`


  version: "LTS - 1.0.0"
schemes:
  - http
  - https
basePath: /v1
produces:
  - application/json
paths:
  /authentication:
    post:
      summary: Authenticate user
      description: >
         Use this endpoint to authenticate a registered user. A successful response returns a JSON object that includes a JWT. Add
the JWT to header files of other Control Room APIs for authorization.

         >

         **Header file example**:
         `-H X-Authorization: <token>`

         >
          - The **username** is the name of a Control Room user.
          - The **password** is the password for the specified user.
          - The **apiKey** is required to configure Single Sign On (SSO). It can also be used in place of a password for users that
are assigned the [API key generation role](https://docs.automationanywhere.com/csh?context=csh-cloud-control-room-apikey-role).
          >
          >
          >> **Note**: In an Enterprise Control Room with SPNEGO properly configured, users do not need to enter a username and
password to generate a JWT.
          >> **SPNEGO** Authentication API URL example: `https://<your_control_room_url/v1/authentication/SPNEGO`

          > **Task based examples:**

          - [Authentication with **username** and **password**](https://docs.automationanywhere.com/csh?context=csh-cloud-
authenticate-password)
          - [Authentication with **username** and **apiKey**](https://docs.automationanywhere.com/csh?context=csh-cloud-
authenticate-apikey)


      tags:
        - auth
      parameters:
        - in: body
          name: body
          required: true
          schema:
            $ref: '#/definitions/AuthRequest'
      responses:
        200:
          description: This is a JSON object that contains the JWT generated for the registered Control Room User. In addition to
the JWT this response contains details regarding the users roles, permissions, and licenses.
          schema:
            $ref: '#/definitions/AuthResponse'
        400:
          $ref: '#/responses/400'
        401:
          $ref: '#/responses/401'
        500:
          $ref: '#/responses/500'
  /authentication/token:
```

```
    get:
      summary: Verify if a token is valid
      description: >
         Use this endpoint to validate an existing JWT. The token is included in the request URL. There is no request body for this
endpoint. The response is either `"valid": true`(the token is valid) or `"valid": false`(the token is not valid).

        >**Task based example:**

         - [Validate an authentication token](https://docs.automationanywhere.com/csh?context=csh-cloud-validate-token)

      tags:
        - auth
      parameters:
        - name: token
          in: query
          type: string
      responses:
        200:
          description: A successful response can be either true or false.
          schema:
            type: object
            properties:
              valid:
                type: boolean
                description: >
                  'true' - if token is valid, 'false' - otherwise
        500:
          $ref: '#/responses/500'
    post:
      summary: Refresh a valid token
      description: >
         Use this endpoint to refresh a valid token.

        - The default token experation is 20 minutes.

        - The token used for the refresh is expired once a refresh token is generated.

         >> **Note:** The default timeout for tokens is configurable. For more information about authentication methods, see
[Supported authentication methods](https://docs.automationanywhere.com/csh?context=cloud-supported-authentication-methods).

        >

         >**Task based example:**

         - [Refresh an authentication token](https://docs.automationanywhere.com/csh?context=csh-cloud-refresh-authentication-
token)
      tags:
        - auth
      parameters:
        - in: body
          name: body
          required: true
          schema:
            $ref: '#/definitions/RefreshAuthRequest'
      responses:
        200:
          description: This is a JSON object containing the refresh token. In addition to the refresh token this response contains
details regarding the users roles, permissions, and licenses.
          schema:
            $ref: '#/definitions/AuthResponse'
        400:
          $ref: '#/responses/400'
        401:
          $ref: '#/responses/401'
        500:
          $ref: '#/responses/500'
  /authentication/logout:
    post:
      summary: Logout a user and delete the JWT
      description: >
         This endpoint immediately invalidates an access token so that it cannot be used for authentication. The token to be logged
out is passed in a header file with the request.

        >> **Header file example:** `-H X-Authorization <token>`


         >**Task based example:**
```

```
          - [Immediately logout (expire) and authentication token](https://docs.automationanywhere.com/csh?context=csh-cloud-
authentication-logout)

      tags:
        - auth
      parameters:
        - in: header
          name: X-Authorization
          description: Authentication token in X-Authorization header for authenitcated user to logout.
          required: true
          type: string
          pattern: '[0-9a-zA-Z-_]*\.[0-9a-zA-Z-_]*\.[0-9a-zA-Z-_]*'
      responses:
        204:
          description: Successful Logout
        500:
          $ref: '#/responses/500'
  /authentication/app/login:
      post:
        summary: Authenticate app User
        description: >
            This endpoint is a service to service authentication API used by Automation Anywhere internally supported
applications. This API is not supported for use by external users.
        tags:
           - auth
        parameters:
          - in: body
            name: body
            required: true
            schema:
              $ref: '#/definitions/AuthRequest'
        responses:
          200:
            description: Auth object containing token generated for specific user
            schema:
              $ref: '#/definitions/AuthResponse'
          400:
            $ref: '#/responses/400'
          401:
            $ref: '#/responses/401'
          500:
            $ref: '#/responses/500'
definitions:
  RoleDto:
    type: object
    required:
      - name
    properties:
      id:
        type: number
        description: Unique identifier representing a specific role
      name:
        type: string
        description: Name of role
  PermissionDto:
    type: object
    properties:
      id:
        type: number
        description: Unique identifier representing a specific permission
      action:
        type: string
      resourceId:
        type: string
      resourceType:
        type: string
  LicenseFeature:
    type: string
    description: License feature
    enum:
      - DEVELOPMENT
      - RUNTIME
      - METABOTRUNTIME
      - IQBOTRUNTIME
  AuthRequest:
    type: object
    properties:
      username:
        type: string
```

```
        description: username of non-AD User
      password:
        type: string
        description: Password of non-AD User
      apiKey:
        type: string
        description: apiKey can be used as a password for any authentication type
  UserDetails:
    type: object
    properties:
      id:
        type: number
        description: Unique identifier representing a specific permission
      roles:
        type: array
        items:
          $ref: '#/definitions/RoleDto'
      permissions:
        type: array
        items:
          $ref: '#/definitions/PermissionDto'
      licenseFeatures:
        type: array
        items:
          $ref: '#/definitions/LicenseFeature'
      principalId:
        type: number
      domain:
        type: string
        description: ActiveDirectory(LDAP) domain
      email:
        type: string
      emailVerified:
        type: boolean
      passwordExpired:
        type: boolean
      passwordSet:
        type: boolean
      enableAutoLogin:
        type: boolean
      username:
        type: string
        description: User login
      firstName:
        type: string
      lastName:
        type: string
      locked:
        type: boolean
  AuthResponse:
    type: object
    properties:
      token:
        type: string
        description: JWT token
      user:
        $ref: '#/definitions/UserDetails'
  RefreshAuthRequest:
    type: object
    properties:
      token:
        type: string
        description: current JWT token
  Error:
    type: object
    properties:
      code:
        type: integer
        description: Error code
      message:
        type: string
        description: Message describing error
responses:
  400:
    description: Bad Rquest
    schema:
      $ref: '#/definitions/Error'
  401:
    description: Authentication required
```

```
  schema:
    $ref: '#/definitions/Error'
500:
  description: Internal Server error
  schema:
    $ref: '#/definitions/Error'
```