1. When server starts, server generates an asymmetrical key pair (RSA)
2. When client starts
   1. Connect to server to acquire server public key (to encrypt client AES key)
   2. Client generates asymmetrical key pair (RSA) (to digital signature request)
   3. Client generates AES key ( to encrypt data body )
3. When client logins
   1. User info is sent to server: user email, password (MD5), client public key
   2. User info is contain in body and the body is encrypted with AES key
   3. AES key is encrypted with server public key and is put in header
4. When server receives login request
   1. Retrieve encrypted AES key in header
   2. Decrypt AES key with server private key
   3. Decrypt body content with AES key
   4. Validate user login info against account table in database
5. If login succeed
   1. Server will store client public key (for future signature verification)
   2. Server returns user id info which is encrypted with AES key (4.b)
6. Afterwards, all the request and response will route the following steps:
   1. Client prepares request body info (for example: JSON format)
   2. Client sign request info (timestamp, user id, content MD5, request URI, request method) with client private key and put the signature in header
   3. Request body is encrypted with AES key
   4. AES key is encrypted with server public key
   5. Timestamp, user id, encrypted AES key are put in header
   6. Client sends request to server
   7. Server retrieves encrypted AES key from header and decrypts it with server private key
   8. Server retrieves body info and decrypts it with AES key (from 6.g)
   9. Server retrieves user id, timestamp info from header and verifies digital signature with related client public key stored in database (5.a)
   10. If signature verifies, then access required resources
   11. After server operation, the response data is encrypted with client AES key (6.g)
7. When client logoff,
   1. Client public key stored in database is cleared
8. All the authentication operations are completed in a customized Filter