**Basic Authentication with PHP and JWT**

When developing application, regardless of web app or windows or mobile app, we will need to implement the authentication for an account.

There are always many approaches and solution out there, and base-auth is one of them.

In this article, I will share my exprerience about the implementation of this base-auth in my project.

* Server script: PHP
* Client: Web/Android App
* Language: Javascript, Java, Kotlin

**Operating Principle**

1. Client sends a request to login with there account to server, with infomation
   * {“userid”:”1234”, “pwd”:”Bg@-|}+Gxw1%#l” }
2. Server recieves the request, then process:

* Access the database to check for user account info.
* If not found, return client with a null token {“token”:null}, in base64 format e+KAnHRva2Vu4oCdOm51bGx9
* If found, use JWT to create a token for the scope requested, with information will be available to client:
  + payload: id, nbf, exp
  + sign the payload with the SERVER\_KEY
* Return the token to client right after done.

1. Client handle token and send database requests:
   * Client check if the token is valid.
   * If invalid, have to show login again or exit depending on user’s expectation.
   * If token is valid, by checking:
     + Base64 decode the token for payload and signature.
     + At the moment, we simply take care of payload.
   * Having payload decoded in a json format with all neccesary info:
     + id, nbf, exp
     + Other information depending on server respond.
   * Before requesting any request to server, client have to check **nbf** and **exp** values, or it will be easily rejected by sever with an error message such as unauthenticated, not logged in, expiration, invalid request…
   * When **sending any request to server** after being authenticated, client has to **always attach the token**, or your request will be rejected as unauthorized right then.
2. How server handle client requests?
   * Perform authentication process for the login
   * Send the token with permission, scope to client
   * Always check the token sending from client and verfiy them before performing any other actions.
   * If not valid, simply return the empty or error response to client
   * If valid, perform the task normally and return the data as user expected.
3. Always keep “userid” or “idno” in every database operation, especially add/update/delete ones, which are always imortant.
4. Implement a BACKUP database
   * This database will track every operation to database via the Server API.
   * Along with it, data of its action will be cloned/back up to be used later when needed.

**Sample source code files**

* server
  + jwt.core.php : core functions working with JWT
  + jwt.php : Wrapper for important function of server app
  + auth.php : Authentication module, handle the auth process
  + tokencheck.php : perform first roughly checking token
  + server.php: sample php file to handle all client request
  + prf.db.php : simulate user profile data
* With these

**Call to authen:**

* <http://localhost/server/auth.php?a=auth&idno=1&userid=john@gmail.com&pwd=123>
* {"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyaWQiOiJqb2huQGdtYWlsLmNvbSIsImlkbm8iOiIxIiwibmJmIjoxNjUyOTQ3ODg2LCJleHAiOjE2NTI5NDg0ODZ9.4a1sg6z\_2nyIp6wGgjXg54qLceijnCwq482WCJZIIyk","exp":1652948486,"config":""}

**Call to fetch info:**

* [http://localhost/server/server.php?a=info&idno=1&userid=john@gmail.com&token=[token](http://localhost/server/server.php?a=info&idno=1&userid=john@gmail.com&token=%5btoken)]
* { "Id":1, "Name" : "John", "Email" : "john@gmail.com" }

**Call to save info:**

* [http://localhost/server/server.php?a=save&idno=1&userid=john@gmail.com&token=[token]&msg=Tiny](http://localhost/server/server.php?a=save&idno=1&userid=john@gmail.com&token=%5btoken%5d&msg=Tiny)
* {"result":"success"}

Try to value of **idno**, **userid** in **fetch** and **save** to see the error.

* {"total\_pages":0,"total\_rows":0,"rows":[],"info":"rejected"}

Hope this helps.