**Documentation for middlewareController.js**

**Overview**

This code file contains the implementation of the middlewareController function, which serves as middleware to authenticate user requests in a web application. It verifies the presence and validity of an authentication token in the request cookies, and allows the request to proceed if the token is valid and the user's account status is active.

**Function: middlewareController**

This function verifies the presence and validity of an authentication token in the request cookies, and allows the request to proceed if the token is valid and the user's account status is active. If the token is missing or invalid, it redirects the user to the login page or ends the response if the token is invalid. If an error occurs during the verification process, it redirects the user to the login page as well.

**Parameters**

* req (object): The request object containing information about the incoming HTTP request.
* res (object): The response object used to send the HTTP response back to the client.
* next (function): The callback function to be invoked if the token is valid and the user's account status is active.

**Returns**

This function does not have a return value.

*const* middlewareController = *async*(*req*, *res*, *next*) *=>* {

    // xác nhận token

*const* auth = *req*.cookies.auth;

    if (!auth) return *res*.redirect("/login");

    try {

*const* [rows] = await connection.execute('SELECT `token`, `status` FROM `users` WHERE `token` = ? AND `veri` = 1', [auth]);

        if(!rows) {

*res*.clearCookie("auth");

            return *res*.end();

        };

        if (auth == rows[0].token && rows[0].status == '1') {

            next();

        } else {

            return *res*.redirect("/login");

        }

    } catch (error) {

        return *res*.redirect("/login");

    }

}

The given code defines a middleware function named middlewareController that checks if a token is present in the request cookies.

1. If the token is missing, it redirects the user to the login page. It then tries to execute a SELECT query to retrieve user details based on the provided token. If the query does not return any rows, it clears the 'auth' cookie and ends the response.
2. If the token matches the one in the database and the user status is '1', it calls the next middleware function. Otherwise, it redirects to the login page. If an error occurs during the process, it also redirects to the login page.