HTTP_Authentication_Services

Application service module that facilitates authentication into the API.

Syntax

```
Response = HTTP_Authentication_Services(@Service, @Params)
```

Returns

The meaning of the response value depends on the service.

Parameters

Parameter	Description
@Service	The name of the service being requested. **Required.**
@Params	Generic parameters. Refer to a specific service to determine the actual parameters used.

Remarks

`HTTP_Authentication_Services` is an application service module that handles authentication into the API. By default this service is automatically called within the HTTP_MCP controller routine. It is recommended that this be left in place. If the developer wishes to disable authentication (either temporarily for testing purposes or permanently), it is better to set the *Enable Authentication Flag* to 0 in the SRP_HTTP_FRAMEWORK_SETUP configuration record.

Authentication is different from authorization (even though *HTTP Basic Authentication* uses the *Authorization* request header field) and this is important when building access to the API. Authentication normally means the user's credentials are valid. However, this does not guarantee that the user has sufficient privileges to access (or modify) the requested resource. This is where authorization comes in. Failure in authentication or authorization can both return a 401 (*Unauthorized*) status code, but a failed authorization might return a 403 (*Forbidden*) status code. Web service code that easily retrieve the authentication credentials to make authorization decisions.

As noted in the comments, HTTP_Authentication_Services supports HTTP Basic Authentication, which is an easy method to implement and is reasonably secure when used over SSL. As noted from the linked Wikipedia article, the rules for using HTTP Basic Authentication are as follows:

> *1. The username and password are combined into a string separated by a colon, e.g.: username:password*
>
> *2. The resulting string is encoded using the RFC2045-MIME variant of Base64, except not limited to 76 char/line.*
>
> *3. The authorization method and a space i.e. "Basic " is then put before the encoded string.*
>
> *For example, if the user agent uses Aladdin as the username and OpenSesame as the password then the field is formed as follows:*
> `Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l`

Services

Service	Description
AuthenticateRequest	**Usage:** `HTTP_Authentication_Services('AuthenticateRequest')` **Comments:** Returns a boolean value indicating the success of the authentication attempt. Default method is built around HTTP Basic Authentication. **Returns:** A boolean value indicating the success of the authentication attempt.
CleanUp	**Usage:** `HTTP_Authentication_Services('CleanUp')` **Comments:** Runs any clean up processes as needed to prepare the engine for the next request. **Returns:** N/A

Params

The proper use of the generic arguments are defined in the definition of each service above.