Output of EnrollUserFactor :

A black background with green lines

Description automatically generated

ActivateFactor :

A computer screen shot of text

Description automatically generated

**VerifyFactor :**

1. This method is an HTTP POST endpoint defined by the [HttpPost("verify")] attribute.
2. It calls the VerifyOtp method from the MfaService to perform the OTP verification.

3.If the verification is successful (the result is not null), it returns an OK response with the verification result.

4.If the verification is not successful (result is null), it returns null (this might need further handling in your specific use case).

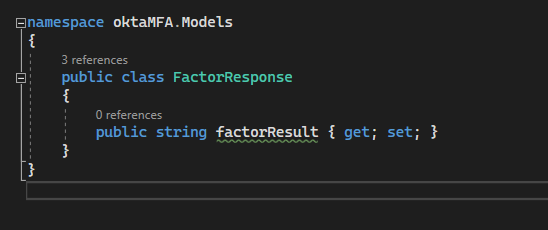
5.It catches HttpRequestException specifically, which might occur if there's an issue with the HTTP request (e.g., network issues).

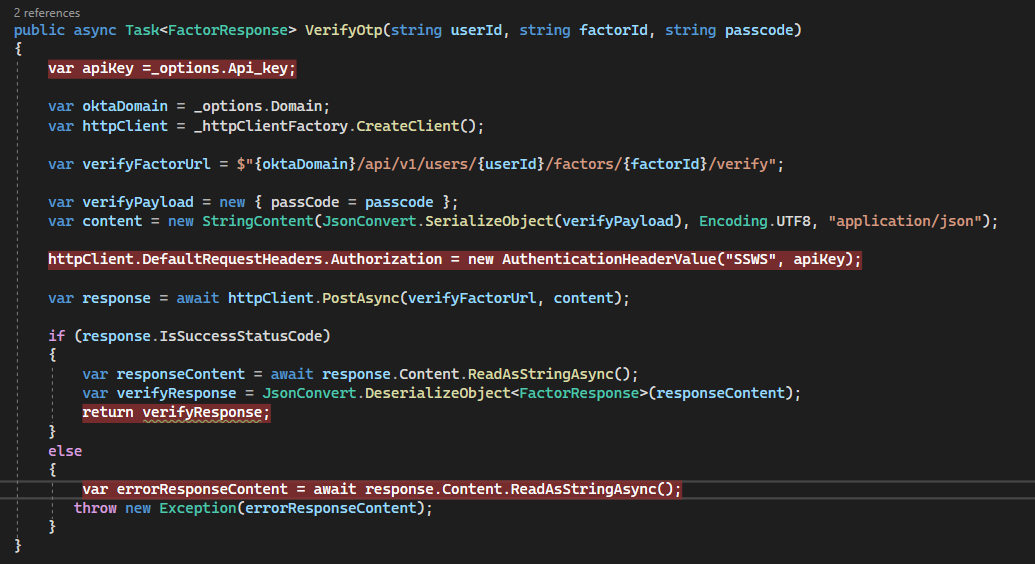
For any other exception, a generic exception catch block is included, returning a BadRequest response with the error message.

A screen shot of a computer program

Description automatically generated

VeriFyOTP :





1.The method starts by retrieving the Okta API key and domain from the \_options object.

It creates an HttpClient to make HTTP requests.

2.The URL for verifying the OTP is constructed using the provided userId and factorId.

3.The passcode is encapsulated in a payload (verifyPayload), which is then converted to JSON and wrapped in a StringContent object.

4.The Okta API key is set in the authorization header of the HTTP client.

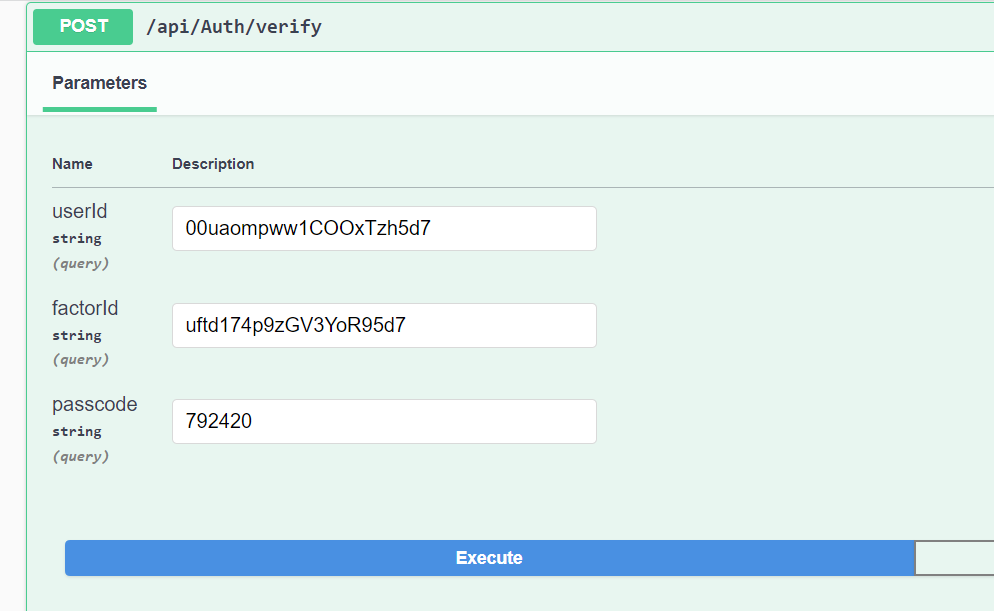
5.POST Request:

A POST request is made to the Okta endpoint for verifying the OTP (verifyFactorUrl), with the passcode payload.

Response Handling:

If the request is successful (status code 2xx), the response content is deserialized into a FactorResponse object using JSON.NET (JsonConvert).

If the request is unsuccessful, an exception is thrown, including the error message from the response content.



A screenshot of a computer

Description automatically generated