

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp-net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/in-aspnet-core-rest-api>)

REST API Model Validation

(<https://www.pragimtech.com/blog/api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Delete in REST API

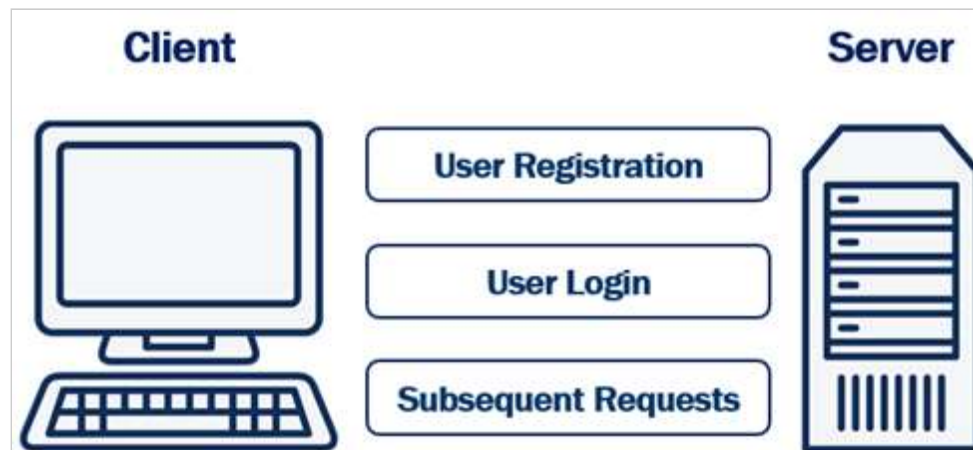
(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Search in REST API

(<https://www.pragimtech.com/blog>)

# How cookie authentication works

In this video we will discuss how cookie authentication works in general. In our upcoming videos we will implement it in Blazor web application.



## User Registration

To be able to login a user needs a user account with our system. So the user fills registration form with their preferred username and password and posts the form to the server. The server then hashes the password and stores it in the database. Hashing prevents password theft. Even if an attacker cracks and gains access to your system, they won't be able to steal passwords because they are hashed. Hashing is different from encryption.

in-asp.net-core-rest-api)

Call REST API from Blazor

are-restful-apis)

Create REST API

(https://www.pragimtech.com/blog/

asp.net-core-rest-api)

REST API DbContext

(https://www.pragimtech.com/blog/

core-rest-api-dbcontext)

Repository Pattern in REST API

(https://www.pragimtech.com/blog/

api-repository-pattern)

REST API Get List

(https://www.pragimtech.com/blog/

core-rest-api-list-of-resources)

REST API Get By Id

(https://www.pragimtech.com/blog/

core-rest-api-get-by-id)

Post in REST API

(https://www.pragimtech.com/blog/

in-aspnet-core-rest-api)

REST API Model Validation

(https://www.pragimtech.com/blog/

api-model-validation)

Put in REST API

(https://www.pragimtech.com/blog/

in-asp.net-core-rest-api)

Delete in REST API

(https://www.pragimtech.com/blog/

in-asp.net-core-rest-api)

Search in REST API

(https://www.pragimtech.com/blog/

## Encryption vs Hashing

MỚI	MỚI	-25%	MỚI	-21%
-24%	-30%	MỚI	MỚI	-15%

Phong Vũ

Encryption is reversible, i.e what is encrypted can also be decrypted. Hashing, is one-way. It is irreversible. Hashing scrambles plain text to produce a unique message digest. If implemented using a strong algorithm, there is no way to reverse the hashing process to reveal the original password.

## Hashing without Salt

Password		Hash	
	<b>Password_1</b>		<b>9gdh3kkj3k3rtqbasua..</b>
	<b>Password_1</b>		<b>9gdh3kkj3k3rtqbasua..</b>

in-asp.net-core-rest-api)  
Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/in-aspnet-core-rest-api>)

REST API Model Validation

(<https://www.pragimtech.com/blog/api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Delete in REST API

(<https://www.pragimtech.com/blog/in-asp.net-core-rest-api>)

Search in REST API

(<https://www.pragimtech.com/blog>)

An attacker can hash random passwords and then compare the hashes to crack the password.

Hashing with Salt

Password		Salt		Hash
Password_1	+	1g3k7t	=	9gdh3kkj3k3rtqbasua..
Password_1	+	2h4p9w	=	8klj5rka2n9trpavdvq..

When a random salt is added to the hashing process, the generated hash will not be the same, even if the plain text passwords are.

in-asp.net-core-rest-api)  
Call REST API from Blazor

are-restful-apis)

Create REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API Model Validation  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Delete in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

Search in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

## User Login



Phong Vũ

The registered username and password can then be used on the login form. The login form is posted to the server. The server looks up the username in the database. Hashes the supplied password, and compares it to the already hashed password in the database. If they match, then the system knows, the user is who he claims to be, otherwise access is denied by sending HTTP status code 401.

If the supplied username and password matches, the server creates an access token which uniquely identifies the user's session. This access token is stored in the database and is also attached to the response cookie. This cookie is then returned to the client. The user is now logged in.

in-asp.net-core-rest-api)  
Call REST API from Blazor

are-restful-apis)

Create REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

REST API Model Validation  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

Delete in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

Search in REST API  
(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)

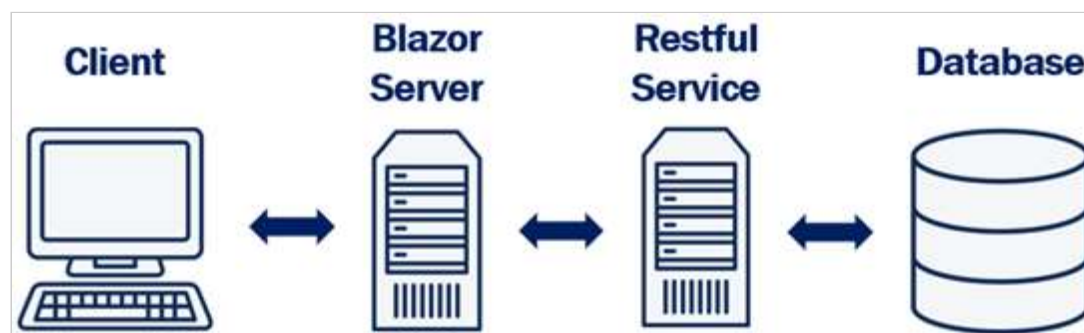
## Subsequent Requests

On every subsequent request, the browser automatically sends the cookie to the server. The server reads the access token from the cookie and checks it against the one in the database associated with that user. If they match, access is granted.

Once the user logs out of the application, both, the authentication cookie and the access token in the database are deleted.

## Cookie authentication or token authentication

The following is our Blazor application architecture. We have a Blazor web application and a Web API. Blazor web application calls WEB API.



- Depending on how you want your application to scale you may have both blazor web app and web api deployed on same server or different servers. Depending

in-asp.net-core-rest-api)  
Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

REST API DbContext

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

REST API Get List

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

REST API Get By Id

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

Post in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

REST API Model Validation

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

Put in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

Delete in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

Search in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)

on the demand, if you want to be able to independently scale up and down these 2 applications, then you may have to deploy them on different servers.

- If they are deployed on different servers, we cannot use the same cookie authentication to authenticate both the Blazor web application and web api. This is because a cookie created by one domain cannot be accessed by another domain.
- Although it is possible to share cookies between sub-domains, it is a standard practice to use cookie based authentication for web applications and token based authentication for web apis.

In our upcoming videos, we will implement cookie authentication to protect our blazor web application and token authentication to protect our Web API.



(<https://www.pragimtech.com/blog/blazor/asp.net-core-identity-setup-in-blazor-application>)  
(<https://www.pragimtech.com/blog/blazor/blazor-cookie-authentication>)



© 2020 Pragimtech (<https://www.pragimtech.com>). All Rights Reserved.



in-asp.net-core-rest-api)

Call REST API from Blazor

are-restful-apis)

Create REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api>)

asp.net-core-rest-api)

REST API DbContext

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-dbcontext>)

core-rest-api-dbcontext)

Repository Pattern in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-repository-pattern>)

api-repository-pattern)

REST API Get List

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-list-of-resources>)

core-rest-api-list-of-resources)

REST API Get By Id

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-get-by-id>)

core-rest-api-get-by-id)

Post in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-post>)

in-aspnet-core-rest-api)

REST API Model Validation

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-model-validation>)

api-model-validation)

Put in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-put>)

in-asp.net-core-rest-api)

Delete in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-delete>)

in-asp.net-core-rest-api)

Search in REST API

(<https://www.pragimtech.com/blog/asp.net-core-rest-api-search>)



Adobe Creative Cloud

**Students save 60%**

Create logos,  
infographics and more



Buy now