⚠ You signed in with another tab or window. Reload to refresh your session.

Join GitHub today

GitHub is home to over 36 million developers working together to host and
review code, manage projects, and build software together.

Sign up

Dismiss

The administration for the IdentityServer4 and Asp.Net Core Identity

#identityserver4 #asp-net-core #aspnetcore-mvc #asp-net-core-identity #identityserver4-admin #identityserver4-adminui #admin #ui

⊙ **591** commits	⑂ **8** branches	♡ **7** releases	⚑ **13** contributors	⚖ MIT

Branch: master ▾   New pull request   Find File   Clone or download ▾

👤 **skoruba** Merge pull request #300 from skoruba/dev …		Latest commit 3729ca3 21 days ago
📁 .github	Create FUNDING.yml	21 days ago
📁 docs	Add swagger-ui picture	last month
📁 src	Remove DbContexts project because all files were moved to EF.Shared p...	24 days ago
📁 templates	Update project template to beta7.	21 days ago
📁 tests	Add new project Skoruba.IdentityServer4.Admin.EntityFramework.Shared ...	25 days ago
📄 .gitignore	Modify gitignore - don't ignore folder Dtos/Log/	last year
📄 LICENSE.md	Update to .Net Core 2.1. (#34)	last year
📄 README.md	Add donation section	21 days ago
📄 Skoruba.IdentityServer4.Admin.sln	Add new project Skoruba.IdentityServer4.Admin.EntityFramework.Shared ...	25 days ago

📖 **README.md**

Skoruba.IdentityServer4.Admin

> The administration of the IdentityServer4 and Asp.Net Core Identity

Project Status

![build passing](https://img.shields.io/badge/build-passing-brightgreen.svg) ![Azure Pipelines succeeded](https://img.shields.io/badge/Azure%20Pipelines-succeeded-brightgreen.svg) ![chat on gitter](https://img.shields.io/badge/chat-on%20gitter-brightgreen.svg)

This is currently in **beta version**

The application is written in the **Asp.Net Core MVC - using .NET Core 2.2**

NOTE: Works only with **IdentityServer4 version 2.3.0 and higher** 🚀

Requirements

⚠ You signed in with another tab or window. Reload to refresh your session.

Installation via dotnet new template

- Install the dotnet new template:

```
dotnet new -i Skoruba.IdentityServer4.Admin.Templates::1.0.0-beta7
```

- Create new project:

```
dotnet new skoruba.is4admin --name MyProject --title MyProject --adminrole MyRole --adminclientid MyClientId --admin
```

Project template options:

```
--name: [string value] for project name
--title: [string value] for title and footer of the administration in UI
--adminrole: [string value] for name of admin role, that is used to authorize the administration
--adminclientid: [string value] for client name, that is used in the IdentityServer4 configuration for admin
client
--adminclientsecret: [string value] for client secret, that is used in the IdentityServer4 configuration for
admin client
```

How to configure the Administration - IdentityServer4 and Asp.Net Core Identity

- Follow these steps for setup project to use existing IdentityServer4 and Asp.Net Core Identity

Template uses following list of nuget packages

- Available nuget packages

Running in Visual Studio

- Set Startup projects:
  - Skoruba.IdentityServer4.Admin
  - Skoruba.IdentityServer4.Admin.Api
  - Skoruba.IdentityServer4.STS.Identity

Configuration of Administration for Deployment

- Configuration of Admin for deploy on Azure
- Configuration of Admin on Ubuntu with PostgreSQL database

Administration UI preview

- This administration uses bootstrap 4

- Admin UI

⚠ You signed in with another tab or window. Reload to refresh your session.

Skoruba IdentityServer4
Admin

The administration for the IdentityServer4 and Asp.Net Core Identity

Clients	Identity Resources	Api Resources
🖥	🏷	☁
Manage	Manage	Manage

Persisted Grants	Users	Roles
🔑	👤	🔒
Manage	Manage	Manage

Skoruba IdentityServer4 Admin
© 2018

- Security token service (STS)

Skoruba IdentityServer4   Settings - admin ▼   logout

Skoruba IdentityServer4

Welcome to Skoruba IdentityServer4

Discovery Document	Persisted Grants	My profile
🖥	🔑	👤
Discovery Document	Persisted Grants	My profile

My personal data	Two-factor authentication	Change password
☁	📱	✏
My personal data	Two-factor authentication	Change password

Here are links to the source code repository, ready to use samples.

Skoruba IdentityServer4
© 2018
　　　　　　　　　　　　　　　　　　　　　　　　　　Language: English ▾

- Forms:

Cloning

```
git clone https://github.com/skoruba/IdentityServer4.Admin
```

Installation of the Client Libraries

```
cd src/Skoruba.IdentityServer4.Admin
npm install

cd src/Skoruba.IdentityServer4.STS.Identity
npm install
```

Bundling and Minification

The following Gulp commands are available:

- `gulp fonts` - copy fonts to the `dist` folder
- `gulp styles` - minify CSS, compile SASS to CSS
- `gulp scripts` - bundle and minify JS
- `gulp clean` - remove the `dist` folder
- `gulp build` - run the `styles` and `scripts` tasks

EF Core & Data Access

- The solution uses these `DbContexts`:

  - `AdminIdentityDbContext` : for Asp.Net Core Identity
  - `AdminLogDbContext` : for logging
  - `IdentityServerConfigurationDbContext` : for IdentityServer configuration store
  - `IdentityServerPersistedGrantDbContext` : for IdentityServer operational store

Visual Studio command line (Nuget package manager):

Migrations for Asp.Net Core Identity DbContext:

```
Add-Migration AspNetIdentityDbInit -context AdminIdentityDbContext -output Data/Migrations/Identity
Update-Database -context AdminIdentityDbContext
```

Migrations for Logging DbContext:

```
Add-Migration LoggingDbInit -context AdminLogDbContext -output Data/Migrations/Logging
Update-Database -context AdminLogDbContext
```

Migrations for IdentityServer configuration DbContext:

```
Add-Migration IdentityServerConfigurationDbInit -context IdentityServerConfigurationDbContext -output Data/Migration
Update-Database -context IdentityServerConfigurationDbContext
```

Migrations for IdentityServer persisted grants DbContext:

```
Add-Migration IdentityServerPersistedGrantsDbInit -context IdentityServerPersistedGrantDbContext -output Data/Migrat
Update-Database -context IdentityServerPersistedGrantDbContext
```

Or via `dotnet CLI` :

Migrations for Asp.Net Core Identity DbContext:

```
dotnet ef migrations add AspNetIdentityDbInit -c AdminIdentityDbContext -o Data/Migrations/Identity
dotnet ef database update -c AdminIdentityDbContext
```

Migrations for Logging DbContext:

```
dotnet ef migrations add LoggingDbInit -c AdminLogDbContext -o Data/Migrations/Logging
dotnet ef database update -c AdminLogDbContext
```

Migrations for IdentityServer configuration DbContext:

```
dotnet ef migrations add IdentityServerConfigurationDbInit -c IdentityServerConfigurationDbContext -o Data/Migration
dotnet ef database update -c IdentityServerConfigurationDbContext
```

Migrations for IdentityServer persisted grants DbContext:

```
dotnet ef migrations add IdentityServerPersistedGrantsDbInit -c IdentityServerPersistedGrantDbContext -o Data/Migrat
dotnet ef database update -c IdentityServerPersistedGrantDbContext
```

Migrations are not a part of the repository - they are ignored in `.gitignore` .

We suggest to use seed data:

- In `Program.cs -> Main` , uncomment `DbMigrationHelpers.EnsureSeedData(host)` or use dotnet CLI `dotnet run /seed`
- The `Clients` and `Resources` files in `Configuration/IdentityServer` are the initial data, based on a sample from IdentityServer4

⚠ You signed in with another tab or window. Reload to refresh your session.

Using other database engines - PostgreSQL, SQLite, MySQL etc.

- Follow these steps for setup other database engines

Authentication and Authorization

- Change the specific URLs and names for the IdentityServer and Authentication settings in `Constants/AuthenticationConsts` or `appsettings.json`
- `Constants/AuthorizationConsts.cs` contains configuration of constants connected with authorization - definition of the default name of admin policy
- In the controllers is used the policy which name is stored in - `AuthorizationConsts.AdministrationPolicy`. In the policy - `AuthorizationConsts.AdministrationPolicy` is defined required role stored in - `AuthorizationConsts.AdministrationRole`.
- With the default configuration, it is necessary to configure and run instance of IdentityServer4. It is possible to use initial migration for creating the client as it mentioned above

Login Configuration

- In `Skoruba.IdentityServer4.STS.Identity` - in `appsettings.json` is possible to specify which column will be used for login (`Username` or `Email`):

  ```
  "LoginConfiguration": {
    "ResolutionPolicy": "Username"
  }
  ```

  or using `Email`:

  ```
  "LoginConfiguration": {
    "ResolutionPolicy": "Email"
  }
  ```

Register Configuration

- In `Skoruba.IdentityServer4.STS.Identity` - in `appsettings.json` is possible to disable user registration (`default: true`):

  ```
  "RegisterConfiguration": {
    "Enabled": false
  }
  ```

How to configure API & Swagger

- For development is running on url - `http://localhost:5001` and swagger UI is available on url - `http://localhost:5001/swagger`
- For swagger UI is configured a client and an API in STS:

  ```
  "AdminApiConfiguration": {
    "IdentityServerBaseUrl": "http://localhost:5000",
    "OidcSwaggerUIClientId": "skoruba_identity_admin_api_swaggerui",
    "OidcApiName": "skoruba_identity_admin_api"
  }
  ```

- Swagger UI contains following endpoints:

⚠ You signed in with another tab or window. Reload to refresh your session.

Skoruba IdentityServer4 Admin Api [v1]
/swagger/v1/swagger.json

Authorize 🔒

ApiResources	⟩
Clients	⟩
IdentityResources	⟩
PersistedGrants	⟩
Roles	⟩
Users	⟩

How to configure an external provider in STS

- In `Skoruba.IdentityServer4.STS.Identity/Helpers/StartupHelpers.cs` - is method called `AddExternalProviders` which contains the example with `GitHub` and in `appsettings.json` :

```
"ExternalProvidersConfiguration": {
        "UseGitHubProvider": false,
        "GitHubClientId": "",
        "GitHubClientSecret": ""
}
```

- It is possible to extend `ExternalProvidersConfiguration` with another configuration properties.

List of external providers for ASP.NET Core:

- https://github.com/aspnet-contrib/AspNet.Security.OAuth.Providers
- https://docs.microsoft.com/en-us/aspnet/core/security/authentication/social/

Azure AD

- Great article how to set up Azure AD:
  - https://azure.microsoft.com/cs-cz/resources/samples/active-directory-dotnet-webapp-openidconnect-aspnetcore/

Email service

- It is possible to set up emails via:

SendGrid

In STS project - in `appsettings.json` :

```
        "SourceEmail": "",
        "SourceName": ""
    }
```

SMTP

```
"SmtpConfiguration": {
        "Host": "",
        "Login": "",
        "Password": ""
    }
```

Localizations - labels, messages

- The project has following translations:
  - English
  - Chinese
  - Russian
  - Persian
  - Swedish

Feel free to send a PR with your translation. 😊

- All labels and messages are stored in the resources `.resx` - locatated in `/Resources`

  - Client label descriptions from - http://docs.identityserver.io/en/latest/reference/client.html
  - Api Resource label descriptions from - http://docs.identityserver.io/en/latest/reference/api_resource.html
  - Identity Resource label descriptions from - http://docs.identityserver.io/en/latest/reference/identity_resource.html

Tests

- The solution contains unit and integration tests.

- **Stage environment is used for integration tests**:

  - `DbContext` contains setup for InMemory database
  - `Authentication` is setup for `CookieAuthentication` - with fake login url only for testing purpose
  - `AuthenticatedTestRequestMiddleware` - middleware for testing of authentication.

- If you want to use `Stage environment` for deploying - it is necessary to change these settings in `StartupHelpers.cs`.

## Overview

Solution structure:

- STS:

  - `Skoruba.IdentityServer4.STS.Identity` - project that contains the instance of IdentityServer4 and combine these samples - Quickstart UI for the IdentityServer4 with Asp.Net Core Identity and EF Core storage and damienbod - IdentityServer4 and Identity template

- Admin UI Api:

  - `Skoruba.IdentityServer4.Admin.Api` - project with Api for managing data of IdentityServer4 and Asp.Net Core

- Admin UI:

    - `Skoruba.IdentityServer4.Admin` - ASP.NET Core MVC application that contains Admin UI

    - `Skoruba.IdentityServer4.Admin.BusinessLogic` - project that contains Dtos, Repositories, Services and Mappers for the IdentityServer4

    - `Skoruba.IdentityServer4.Admin.BusinessLogic.Identity` - project that contains Dtos, Repositories, Services and Mappers for the Asp.Net Core Identity

    - `Skoruba.IdentityServer4.Admin.BusinessLogic.Shared` - project that contains shared Dtos and ExceptionHandling for the Business Logic layer of the IdentityServer4 and Asp.Net Core Identity

    - `Skoruba.IdentityServer4.Admin.EntityFramework` - EF Core data layer that contains Entities for the IdentityServer4

    - `Skoruba.IdentityServer4.Admin.EntityFramework.Identity` - EF Core data layer that contains Repositories for the Asp.Net Core Identity

    - `Skoruba.IdentityServer4.Admin.EntityFramework.Extensions` - project that contains extensions related to EntityFramework

    - `Skoruba.IdentityServer4.Admin.EntityFramework.Shared` - project that contains DbContexts for the IdentityServer4, Logging and Asp.Net Core Identity, inluding shared Identity entities

- Tests:

    - `Skoruba.IdentityServer4.Admin.IntegrationTests` - xUnit project that contains the integration tests for AdminUI

    - `Skoruba.IdentityServer4.Admin.UnitTests` - xUnit project that contains the unit tests for AdminUI

    - `Skoruba.IdentityServer4.STS.IntegrationTests` - xUnit project that contains the integration tests for STS

The admininistration contains the following sections:
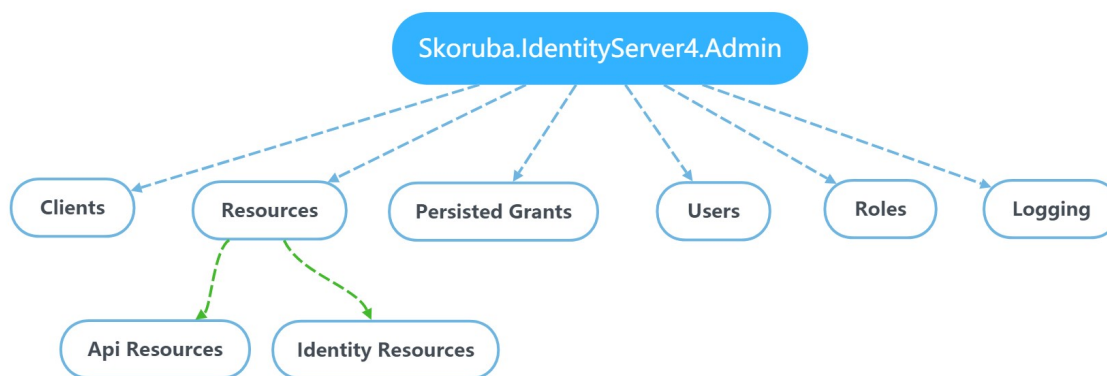
IdentityServer4

Clients

It is possible to define the configuration according the client type - by default the client types are used:

- Empty

- Web Application - Server side - Hybrid flow

- Native Application - Mobile/Desktop - Hybrid flow

- Machine/Robot - Resource Owner Password and Client Credentials flow

- TV and Limited-Input Device Application - Device flow

- Actions: Add, Update, Clone, Remove

- Entities:

   - Client Cors Origins
   - Client Grant Types
   - Client IdP Restrictions
   - Client Post Logout Redirect Uris
   - Client Properties
   - Client Redirect Uris
   - Client Scopes
   - Client Secrets

API Resources

- Actions: Add, Update, Remove
- Entities:
   - Api Claims
   - Api Scopes
   - Api Scope Claims
   - Api Secrets
   - Api Properties

Identity Resources

- Actions: Add, Update, Remove
- Entities:
   - Identity Claims
   - Identity Properties

Asp.Net Core Identity

Users

- Actions: Add, Update, Delete
- Entities:
   - User Roles
   - User Logins
   - User Claims

Roles

- Actions: Add, Update, Delete
- Entities:
   - Role Claims

Application Diagram
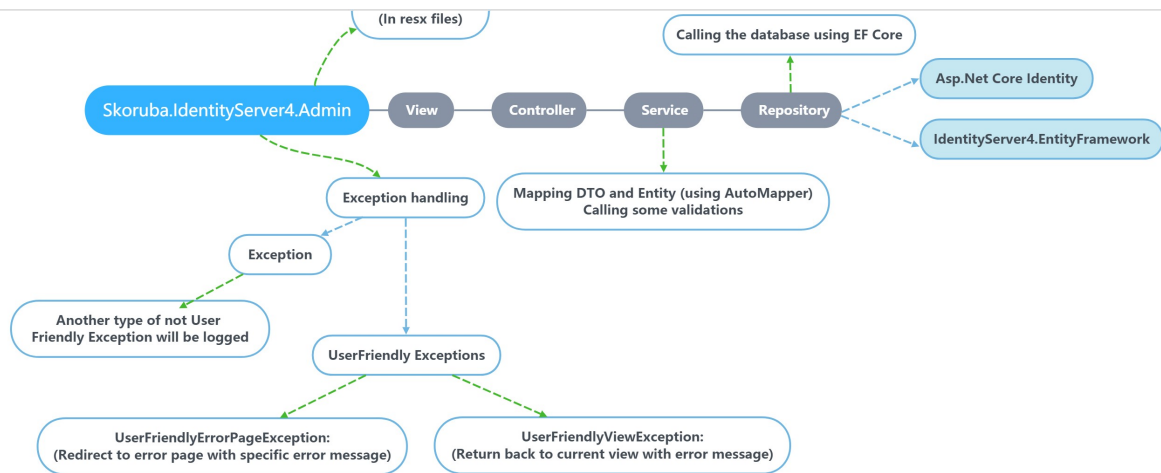
Roadmap & Vision

1.0.0:

- ☑ Create the Business Logic & EF layers - available as a nuget package
- ☑ Create a project template using dotnet CLI - `dotnet new template`
  - ☑ First template: The administration of the IdentityServer4 and Asp.Net Core Identity
- ☑ Add logging into
  - ☑ Database
  - ☑ File
- ☑ Add localization for other languages
  - ☑ English
  - ☑ Chinese
  - ☑ Russian
  - ☑ Persian
  - ☑ Swedish
- ☑ Manage profile
- ☑ Password reset
- ☑ Link account to an external provider (example with Github)
- ☑ Two-Factor Authentication (2FA)
- ☑ User registration
- ☑ Email service
  - ☑ SendGrid
- ☑ Add API
  - ☑ IdentityServer4
  - ☑ Asp.Net Core Identity
  - ☑ Add swagger support

1.1.0

- ☐ Add audit logs to track changes (#61)

2.0.0:

- ☐ Docker support (#121)
- ☐ Create a project template using dotnet CLI - `dotnet new template`

⚠ You signed in with another tab or window. Reload to refresh your session.

Future:

- Add UI tests (#97, #116)
- Add more unit and integration tests 😊
- Extend administration for another protocols
- Create separate UI using `Razor Class Library` (#28, #133)

Licence

This repository is licensed under the terms of the **MIT license**.

NOTE: This repository uses the source code from https://github.com/IdentityServer/IdentityServer4.Quickstart.UI which is under the terms of the **Apache License 2.0**.

Acknowledgements
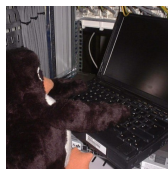
This web application is based on these projects:
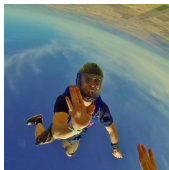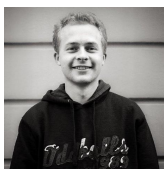
- ASP.NET Core
- IdentityServer4.EntityFramework
- ASP.NET Core Identity
- XUnit
- Fluent Assertions
- Bogus
- AutoMapper
- Serilog

Thanks to Tomáš Hübelbauer for the initial code review.

Thanks to Dominick Baier and Brock Allen - the creators of IdentityServer4.

Contributors

Thanks goes to these wonderful people (emoji key):

Jan Škoruba 💻 💬 📖 💡 🤔	**Tomáš Hübelbauer** 💻 👀 📖 🤔	**Michał Drzał** 💻 👀 📖 💡 🤔	**cerginio** 💻 🐛 💡 🤔	**Sven Dummis** 📖	**Seaear** 💻 🌐
Rune Antonsen 🐛	**Sindre Njøsen** 💻	**Alevtina Brown** 🌐	**Brice** 💻	**TheEvilPenguin** 💻	**Saeed Rahmani** 🌐

⚠ You signed in with another tab or window. Reload to refresh your session.
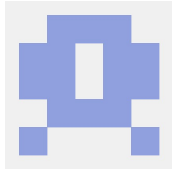
Jan Škoruba 💻 💬 📖 💡 🤔	**Tomáš Hübelbauer** 💻 👀 📖 🤔	**Michał Drzał** 💻 👀 📖 💡 🤔	**cerginio** 💻 🐛 💡 🤔	**Sven Dummis** 📖	**Seaear** 💻 🌍
Andy Yu 🌍					

This project follows the all-contributors specification. Contributions of any kind are welcome!

Contact and Suggestion

I am happy to share my attempt of the implementation of the administration for IdentityServer4 and ASP.NET Core Identity.

Any feedback is welcome - feel free to create an issue or send me an email - jan@skoruba.com. Thank you 😊

Support and Donation 🕊

If you like my work, you can support me by donation. 👍

https://www.paypal.me/skoruba