Progetto Applicazioni Web – JobScheduler

Il progetto è stato realizzato usando ASP.NET Core & Blazor Server. La soluzione è divisa in tre progetti:

JobScheduler

Questo progetto contiene la logica per il Master. Implementa una web app e delle API protette tramite JWT. La documentazione per le API è presente all'indirizzo /swagger.

BackgroundWorker contiene due classi con il compito di schedulare, eseguire e richiedere ad uno slave di eseguire un dato job.

Controllers contiene i controller per le API e delle classi contenenti i metodi esposti dalle API.

JobScheduler.Shared

È un progetto condiviso usato per condividere i modelli fra master e slave.

JobScheduler.Slave

Implementazione dello slave che contiene un set limitato di API per la ricezione di comandi da parte del master (con documentazione all'indirizzo /swagger). Gli indirizzi IP autorizzati sono limitati tramite un middleware che consente solamente indirizzi IP locali.

Istruzioni per il setup iniziale

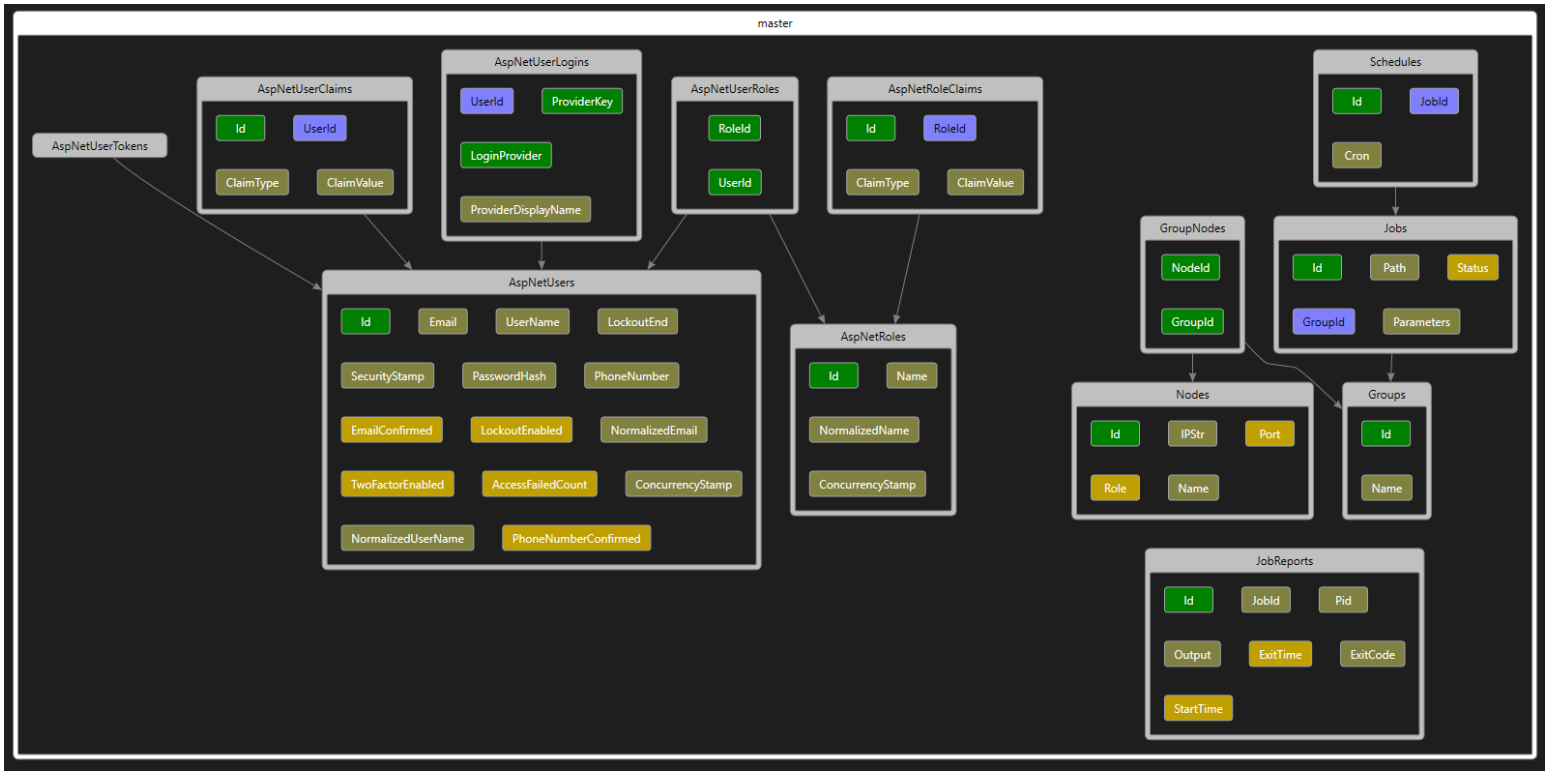
Il progetto contiene un database di test chiamato master.db con tutti i dati già presenti per l'avvio. Sono presenti 2 schedule.

Se non si vuole usare il database già presente, eliminare il file e seguire le seguenti istruzioni:

1. Creare una migrazione (*Add-Migration CreateDB*) e applicarla al database (*Update-Database*).
2. All'avvio del master:
    - Un account admin verrà creato, le credenziali sono:
        - email/username: admin@jobscheduler.com
        - password (case sensitive): P@ssW0rd123
    - Due nodi rappresentanti il master e lo slave verranno creati. Di default entrambi avranno come indirizzo IP localhost e una porta di default.
3. Effettuare il login tramite le credenziali fornite dopodiché:
    - Creare gruppo nella pagina "Groups"
    - Aggiungere i nodi esistenti ad un gruppo tramite il tasto "Edit" nella pagina "Nodes"
    - Creare un Job tramite il tasto "Create Job" nella pagina "Jobs"
    - Schedulare l'esecuzione di un Job tramite il tasto "Create Schedules" nella pagina "Schedules"

È possibile gestire gli utenti tramite la pagina "Administration"

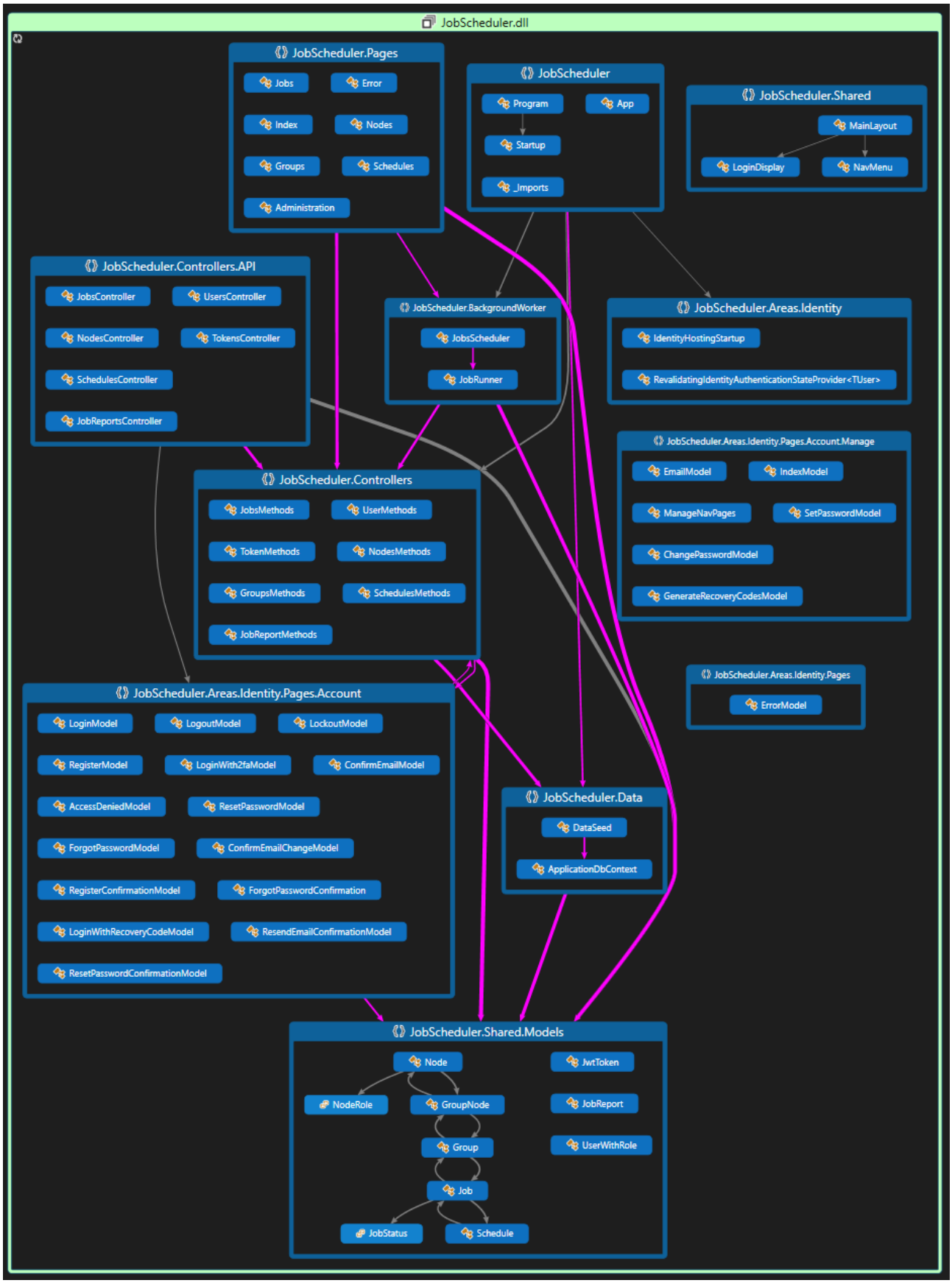
Modello dei dati



- La tabella **Schedules** contiene tutte le informazioni per i job schedulati:
  - **Id**: L'id della schedule
  - **JobId**: L'id del Job oggetto della schedulazione
  - **Cron**: Un'espressione CRON che specifica quando eseguire il job
- **Jobs** contiene le informazioni di un job:
  - **Id**: L'id del job
  - **Path**: Il path del file da eseguire
  - **Parameters**: I parametri da passare al file da eseguire
  - **Status**: Lo stato del Job (eseguito, in esecuzione, in attesa) - non tutte le evenienze sono usate nel programma
  - **GroupId**: L'id del gruppo sul quale eseguire il job
- **Nodes** contiene le informazioni di un nodo:
  - **Id**: L'id del nodo
  - **IPStr**: Una stringa contenente l'IP del nodo
  - **Port**: La porta sul quale il nodo risponde

    - **Role**: Il ruolo del nodo (master/slave)
    - **Name**: Il "nickname" del nodo
- **Groups** contiene le informazioni di un gruppo:
    - **Id**: L'id del gruppo
    - **Name**: Un nome per identificare facilmente il gruppo
- **JobReports** contiene le informazioni su tutti i job eseguiti:
    - **Id**: L'id del report
    - **JobId**: L'id del job eseguito
    - **Pid**: Il PID assegnato dal sistema al job al momento dell'esecuzione
    - **Output**: L'output generato dal job durante l'esecuzione
    - **ExitTime**: Informazioni sull'orario di completamento del job
    - **ExitCode**: L'exitcode del job
    - **StartTime**: L'orario di avvio del job
- Le tabelle AspNet[xxx] sono create da ASP.NET Core Identity

Code map


JobScheduler.dll
JobScheduler.Pages
Jobs
Error
Index
Nodes
Groups
Schedules
Administration
JobScheduler
Program
App
Startup
_Imports
JobScheduler.Shared
MainLayout
LoginDisplay
NavMenu
JobScheduler.Controllers.API
JobsController
UsersController
NodesController
TokensController
SchedulesController
JobReportsController
JobScheduler.BackgroundWorker
JobsScheduler
JobRunner
JobScheduler.Areas.Identity
IdentityHostingStartup
RevalidatingIdentityAuthenticationStateProvider<TUser>
JobScheduler.Areas.Identity.Pages.Account.Manage
EmailModel
IndexModel
ManageNavPages
SetPasswordModel
ChangePasswordModel
GenerateRecoveryCodesModel
JobScheduler.Controllers
JobsMethods
UserMethods
TokenMethods
NodesMethods
GroupsMethods
SchedulesMethods
JobReportMethods
JobScheduler.Areas.Identity.Pages
ErrorModel
JobScheduler.Areas.Identity.Pages.Account
LoginModel
LogoutModel
LockoutModel
RegisterModel
LoginWith2faModel
ConfirmEmailModel
AccessDeniedModel
ResetPasswordModel
ForgotPasswordModel
ConfirmEmailChangeModel
RegisterConfirmationModel
ForgotPasswordConfirmation
LoginWithRecoveryCodeModel
ResendEmailConfirmationModel
ResetPasswordConfirmationModel
JobScheduler.Data
DataSeed
ApplicationDbContext
JobScheduler.Shared.Models
Node
JwtToken
NodeRole
GroupNode
JobReport
Group
UserWithRole
Job
JobStatus
Schedule