ApplicationUser

+ Userld: int

+ UserName: string

+ Email: string

+ FullName: string

+ Age: int

+ HeightInches: int

+ Weight: float

+ Token: string

PreDeterminedWorkout

+ Workoutld: int

+ Title: string

+ DayOfWeek: string

+Exercises: List<int>

FitDeck.Models

ApplicationUserIdentity

- + UserId: int
- + UserName: string
- + NormalizedUserName: string
- + Email: string
- + NormalizedEmail: string
- + FullName: string
- + Age: int
- + HeightInches: int
- + Weight: float
- + PasswordHash: string

ApplicationUserLogin

- + UserName: string
- + Password: string

ApplicationUserRegister

- + Email: string
- + FullName: string
- + Age: int
- + HeightInches: int
- + Weight: float

ApiException

- + StatusCode: int
- + Message: string
- + ToString(): string

PreDeterminedRoutine

- + Routineld: int
- + Title: string
- + RestDay: string
- +Exercises: List<PreDeterminedWorkout>

UserCreateWorkout

- + Workoutld: int
- + UserId: int
- + Title: string
- + DayOfWeek: string
- +Exercises: List<int>

UserCreatedRoutine

- + Routineld: int
- + UserId: int
- + Title: string
- + RestDay: string
- +Exercises: List<PreDeterminedWorkout>

- + _repository: IAccountRepositor
- + CreateAsync(ApplicationUserId
- +FindByNameAsync(string, Cand
- + GetEmailAsync(ApplicationUse
- + GetNormailizedEmailAsync(Ap
- + GetUserId(ApplicationUserIder
- + GetUserNameAsync(Application
- + SetEmailAsync(ApplicationUse
- + SetNormalizedEmailAsync(App
- + SetNormalizedUserNameAsyn
- + SetPasswordHashAsync(Applic

FitDeck.Repository

FitDeck.Identity

UserStore

lentiy, CancellationToken): Task<IdentityResult>

cellationToken): Task<ApplicationUserIdentity>

rldentiy, CancellationToken): Task<string>

plicationUserIdentiy, CancellationToken): Task<string>

tiy, CancellationToken): Task<string>

nUserIdentiy, CancellationToken): Task<string>

rldentiy, string, CancellationToken): Task

olicationUserIdentiy, string, CancellationToken): Task

c(ApplicationUserIdentiy, string, CancellationToken): Task

cationUserIdentiy, string, CancellationToken): Task

Account Account config: IConf - AccountRepo RoutineAsync(nc(): Task <list task<list="" task<list<pre="" vorkout(int):=""> Task<list<pre> Task<list< pre=""></list<></list<pre></list>	
Account _config: IConf - AccountRepo RoutineAsync(nc(): Task <lis: p="" task<="" vorkout(int):=""></lis:>	<in IAccou</in
Account _config: IConf - AccountRepo RoutineAsync(nc(): Task <lis p="" task<="" vorkout(int):=""> Task<list< p=""></list<></lis>	plicationUserIdentity,
Account _config: IConf - AccountRepo RoutineAsync(nc(): Task <lis p="" task<="" vorkout(int):=""> Task<list< p=""></list<></lis>	e/string Cancellation
_config: IConf -AccountRepo RoutineAsync(nc(): Task <lis: task<list<pree<="" td="" vorkout(int):=""><td>string, Cancellation</td></lis:>	string, Cancellation
_config: IConf -AccountRepo RoutineAsync(nc(): Task <lis: task<list<pree<="" td="" vorkout(int):=""><td></td></lis:>	
RoutineAsync(nc(): Task <list< td=""><td>Accoun</td></list<>	Accoun
RoutineAsync(nc(): Task <lis Vorkout(int): Ti ask<list<pred< td=""><td>config: IConfig</td></list<pred<></lis 	config: IConfig
nc(): Task <lis Vorkout(int): Ta ask<list<pree< td=""><td>+ AccountRepos</td></list<pree<></lis 	+ AccountRepos
nc(): Task <lis Vorkout(int): Ta ask<list<pree< td=""><td></td></list<pree<></lis 	
nc(): Task <lis Vorkout(int): Ta ask<list<pree< td=""><td></td></list<pree<></lis 	
nc(): Task <lis Vorkout(int): Ta ask<list<pree< td=""><td>FromRoutineAsync(ii</td></list<pree<></lis 	FromRoutineAsync(ii
āsk <list<pre[< td=""><td>esAsync(): Task<list< td=""></list<></td></list<pre[<>	esAsync(): Task <list< td=""></list<>
mRoutine(int)	FromWorkout(int): Ta its(): Task <list<pred< td=""></list<pred<>
mRoutine/int)	
mRoutine(int)	
mRoutine(int)	
nmRoutine(int)	
mRoutine(int)	
mRoutine(int):	
mRoutine(int):	
	utsFromRoutine(int):
	esByUserId(int): Task FromWorkout(int): Ta
	tsByUserId(int): Tasl
	JserCreatedWork

erface> tRepository

CancellationToken): Task<IdentityResult>

Token): Task<ApplicationUserIdentity>

tRepository

uration

itory(IConfiguration)

<interface> **IPreDeterminedRepository**

t): Task<List<PreDeterminedWorkout>>

PreDeterminedRoutine>>

k<List<int>>

eterminedWorkout>>

PreDeterminedRepository

- -_config: IConfiguration
- + PreDeterminedRepository(IConfiguration)

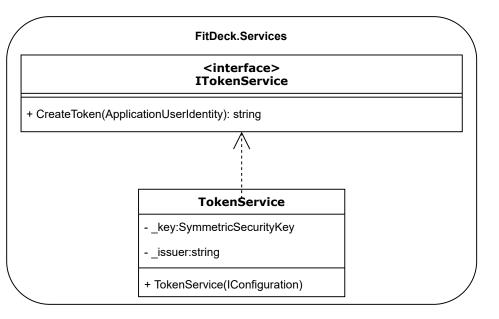
<interface> **IUserCreatedRepository**

Task<List<UserCreatedWorkout>>

<List<UseCreatedRoutine>>

k<List<int>>

<List<UserCreatedWorkouts>> : Task





- _tokenService: ITokenService
- _userManager: UserManager<ApplicationUserIdentity>
- signInManager: SignInManager<ApplicationUserIdentity>
- + Register(ApplicationUserRegister): Task<ActionResult<ApplicationUser>>
- + Login(ApplicationUserLogin): Task<ActionResult<ApplicationUser>>

FitDeck.We

+

nUserIdentiy, string, CancellationToken): Task	
ab	
ExceptionMiddlewareExtension	
ConfigureExceptionHandler(this IApplicationBuilder): void	
Program	
Main(string[]): void	
CreateHostBuilder(string[]): void	
Startup	
config: IConfiguration	
ConfigureServices(IServiceCollection): void	
Configure(IApplicationBuilder, IWebHostEnvironment): void	

-_config +UserCr

