**Web API check list**

1. **Create Project**

* [ProjectName].Models class libary- Add System.ComponentModel.DataAnnotations
* [ProjectName].Data class libary
* [ProjectName].WebApi – Web Api 2 project (with users and tests?)
* [ProjectName].WCF – WCF Application

1. **Users Setup**

* Move User ApplicationUser from WebApi/Models/IdentityModels to [ProjectName].Models -> ProjectNameUser.cs
* Move User ApplicationDbContext from WebApi/Models/IdentityModels to [ProjectName].Data -> ProjectNameDbContext.cs
* Fix references: Add AspNet.Identity.Core to [ProjectName].Data and in WebApi project IdentityConfig add references to [ProjectName].Data and [ProjectName].Models. Also to Startup.Auth AccountsController and ApplicationOAuthProvider
* Rename the user class and dbContext class
* Change password requirements in IdentityConfig

1. **Connection string**

* Add correct connection string to [ProjectName].Data app.config
* Replace the string in [ProjectName].WebApi web.config

<connectionStrings>

<add name="DefaultConnection" connectionString="Data Source=.;Initial Catalog=ProjectNameDb; Integrated Security = true;" providerName="System.Data.SqlClient"/>

</connectionStrings>

* If there are problems with expressions add to connection string multipleResultSets=true;

1. **Create all Models**

* Add System.ComponentModel.DataAnnotations for validations
* ClassNameId and virtual ClassName for FK
* Virtual ICollection<ClassName> and Constructor initialization with Hashset for PK class in FK relationship
* Virtual ICollection<ClassName> in both classes for many to many

1. **Extend [ProjectName]DbContext to contain all the needed db sets**

* Add all needed models db sets
* Add interface
* Add constructors – one empty with default connection string and one with parameter connection string (Add this to Ninject dependency container to use with the parametrized constructur)
* Enable Migrations - package manager > enable\_migrations (for [ProjectName].Data);
* Migrations / Configuration() > this.AutomaticMigrationDataLossAllowed = true;
* Constructur add

1. **Repositories – one for WebApi and one for Wcf**

* Public: All, Find, Add, Update, Delete(T), Delete(id), int SaveChanges
* Private ChangeState(T entity, EntityState state); prop DbContext; IDbSet<T>

1. **UoW – Data classes**

* Base Data Class – private DbContext context , private Dictionary<Type,object> repositories, SaveChanges, Private/protected GetRepository<T>()
* Inherit by concreate [ProjectNam]Data class that implements also properties with get for each repository

1. **BaseControler**

* Create base Controler in WebApi

**BaseController**

protected IArticlesData data;

public BaseApiController(IArticlesData data)

{

this.data = data;

}

1. **InversonOfControl -** Ninject.web.webapi OWIN add to [ProjcetName].WebApi

* GitHub link: <https://github.com/ninject/Ninject.Web.Common/wiki/Setting-up-a-OWIN-WebApi-application>
* Add namespaces:

using Ninject;

using Ninject.Web.Common.OwinHost;

using Ninject.Web.WebApi.OwinHost;

* Add to Startup.cs Configuration

app.UseNinjectMiddleware(CreateKernel)

.UseNinjectWebApi(GlobalConfiguration.Configuration);

* In Startup.cs Create private static CreateKernel method

private static StandardKernel CreateKernel()

{

var kernel = new StandardKernel();

kernel.Load(Assembly.GetExecutingAssembly());

RegisterMappings(kernel);

return kernel;

}

* In Startup.cs Create private static RegisterMappings(kernel) and add for each binding

kernel.Bind<IData>().To<Data>().WithConstructorArgument("context", c => new DbContext());

* Ninject breaks css on home page and Api page loading – in BundeConfig.cs BundleTable.EnableOptimizations = false to fix css.

1. **Cors for WebApi 2.2** – Microsoft ASP.NET Web API 2.2 Cross-origin

* Add in AppStart/WebApiConfig.cs -> config.EnableCors(new EnableCorsAttribute("\*", "\*", "\*"));
* Add in Global.asax >

protected void Application\_BeginRequest(object sender, EventArgs e)

{

Response.Headers.Add("Access-Control-Allow-Origin", "\*");

}

1. **Routing -** WebApiConfig.cs > for routings

* For login – App\_Start/Startu.Auth.cs TokenEndointPath = new PathString([loginRout]);
* In AccountController change [RoutePrefix("api/Users")]

1. **Controlers**

**WCF Check list**

* **[Service].svc**

Factory="System.ServiceModel.Activation.WebServiceHostFactory"

* **[ServiceInterface]**

[WebGet(UriTemplate="")]

* **Web.config**

<system.serviceModel>

<standardEndpoints>

<webHttpEndpoint>

<standardEndpoint helpEnabled="true" defaultOutgoingResponseFormat="Json" />

</webHttpEndpoint>

</standardEndpoints>

</system.serviceModel>