

Sprint 1 – Use Cases & Basic Swagger API

1. Im ersten Sprint geht es um die Use Cases und das Design der Entities:

- Definieren Sie alle Use Cases im User Story Format – als User Stories auf AZURE DEVOPS
- Entwerfen Sie das Domain Model (Aggregates) nach DDD.

Geben Sie das Domain Model als PDF auf Moodle ab.

Format des Dateinames: Nachname1_ Nachname2_ UE1.pdf.

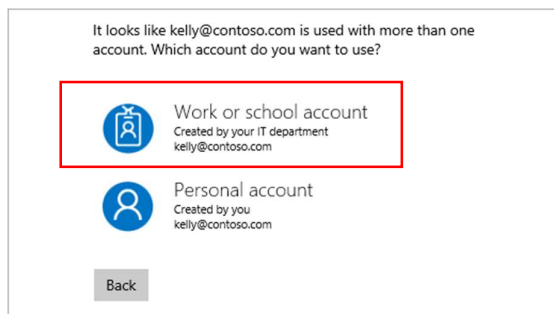
Legen sie die User Stories als Workitems unter „BOARDS / Workitems“ auf Azure DevOps an.



2. Installieren Sie die **Entwicklungsumgebung** Visual Studio Code.
3. Legen Sie das Azure DevOps Konto an (auf <http://dev.azure.com>).

Achtung: Verwenden Sie ausschließlich x@technikum-wien.at Benutzerkonten!

Stellen Sie sicher, dass sie die **FIRMEN/SCHUL** Konten verwenden, nicht etwaige PRIVATE Konten.



Fügen Sie mich (knor@technikum-wien.at) hinzu (<https://aka.ms/Azure DevOpsadduser>).

4. Erweitern Sie die Swagger Contracts (Version 1.0) auf Moddle:

a. Fügen Sie folgende neue Operation hinzu:

The image shows a Swagger UI snippet for a new operation. The operation is a POST request to the endpoint `/parcel/{trackingId}/reportHop/{code}`. The description states: "Report that a `Parcel` has arrived at a certain hop either `warehouse` or `Truck`." The parameters section lists two required path parameters: `trackingId` (type `string`) and `code` (type `string`). The responses section shows a `200` response with the description "Successfully reported hop." and a `500` response with the description "An error occurred." and a schema for an error object: `{ error: { errorMessage: string } }`. A "Try this operation" button is visible at the bottom.

```
POST /parcel/{trackingId}/reportHop/{code}

Description
Report that a Parcel has arrived at a certain hop either warehouse or Truck.

Parameters
Name      Located in  Description                                     Required  Schema
trackingId path      The tracking ID of the parcel.                   Yes       string
code       path      The Code of the hop ( warehouse or Truck ).    Yes       string

Responses
Code      Description                                     Schema
200       Successfully reported hop.
500       An error occurred.                               { error: { errorMessage: string } }
```

Verwenden Sie dazu <http://editor.swagger.io> oder VS Code.

5. Implementieren Sie REST Services mit Hilfe von **ASP.NET WebAPI**

- Verwenden Sie das die Kontrakte von (4).
- Lassen Sie sich mittels <http://editor.swagger.io> die entsprechenden leeren Server ApiController generieren.
- Entfernen Sie `IEquatable`, sowie alle anderen Interfaces der DTOs/Entities/Models und deren Methoden
- Stellen Sie das Framework auf .NET 5 um und passen Sie die NuGet Packages entsprechend für .NET 5 an.

6. Teilen Sie die Solution von (5) in einzelne Projekte auf..

- Ein Projekt für die Webservice Implementierung, eines für die Interfaces, eines für datenhaltende Klasse (DTOs – „Data Transfer Objects“ – aka. Models – aka. Entities).
- Ordnungsgemäße Namespaces (`<Organisation>.<Produktname>.<Bereich>.<Teil>`)



c. **Alle Projekte sollen .NET 5 Projekte sein.**

(Web Site und Class Libraries)

Die Verzeichnisse, Projektnamen, auch von generierten Projekten bitte anpassen!

7. Abgabe:

Der MAIN Branch ihres Azure DevOps Git Repositories sollte immer den Abgabestand des Source Codes enthalten. Arbeiten Sie auf einem anderen Branch und mergen Sie auf MAIN, sobald Sie abgeben / releasen.

Beide GruppenteilnehmerInnen müssen die Abgabe (=Angabe der Daten) auf Moodle machen!