# Background

Application is managing trade orders.

New requirement is to implement ability to link together multiple orders.

A link is described by it's identity (integer field). Following constraints are imposed:

- An order can be part of only a single link

- Only specific trades are allowed to be linked together: having same product symbol and sub-account

## Code details

- `MatchbookDbContext` class in `Matchbook.Db` project is describing data model

- `Matchbook.Model` project is containing the entities

- `Matchbook.WebHost` is ASP.NET Core 3.0 project exposing REST endpoints

- `Order` entity already have a relation to `OrderLink`

- Database is deleted and re-created at each run, fake data is generated to seed the database at startup

## Part 1: Implement order linking REST API

Order linking should be exposed through Web API endpoint (`/orderlinking`) with corresponding

`OrderLinkingController`:

- In case if link creation fails, a proper HTTP code should be returned

together with a message describing the error

- If link creation is successful, a `HTTP 201 Created` status should be returned together with new link id

Tasks:

- Add ability to handle POST request containing a list of order IDs

- Validate request

- Create and save the link, if possible, given constraints mentioned above

- Fail and return proper response if validation failed

## Part 2: Link name

A new requirement is to have a name for a link (string field), which should be unique

Tasks:

- Add missing property to the Link with proper model configuration

- Enforce link name uniqueness with database constraints

- Include link name validation at the controller level

## Optional scope for bonus points

- Tweak data model to improve query performance for link creation scenario

- Unit tests

- Comments or recommendations around application's structure/architecture