[Task Name]

Technical Document

Date: 10/Aug/2018

Version: Version 0.1

# Table of Contents

[Table of Contents 1](#_Toc928258)

[[Task Name] 2](#_Toc928259)

[Description 2](#_Toc928260)

[1.1 Mockup screen: 2](#_Toc928261)

[1.2 Angular & Spring Models 3](#_Toc928262)

[1.3 Angular Component 3](#_Toc928263)

[1.4 Rest Services 3](#_Toc928264)

[1.5 Flow Services: 4](#_Toc928265)

[1.6 Repository Functions 4](#_Toc928266)

[1.7 Database Script: 4](#_Toc928267)

[1.8 ER Diagram: 4](#_Toc928268)

# 

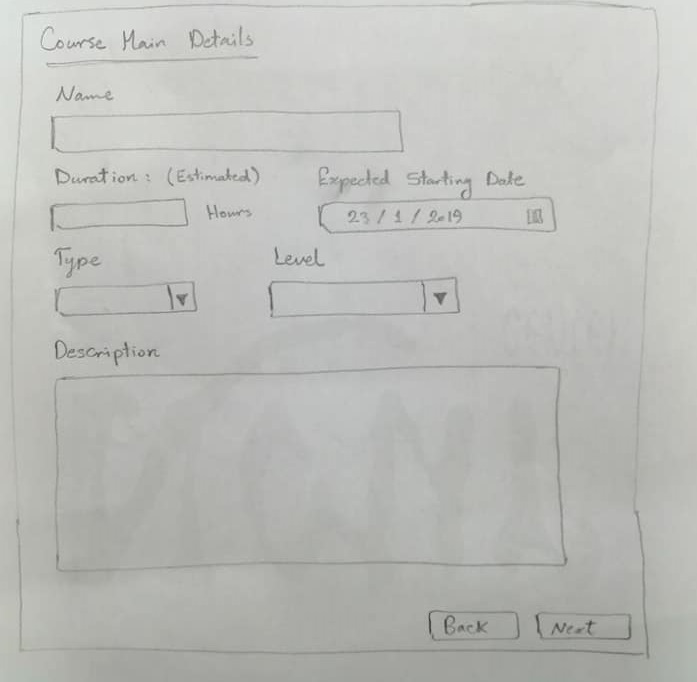
# [Task Name]

Refer to: [Taiga URL]

## Description

Write the Task Description Here

## Mockup screen:



## Angular & Spring Models

|  |  |  |  |
| --- | --- | --- | --- |
| **New** | **Class Name** | CourseDTO | |
| **Angular Path** | | app/modules/course/shared/data/course-dto.data.ts | |
| **Spring Path** | | model/course/CourseDTO | |
| **Fields** | | | |
| **Attribute** | | **Angular Data Type** | **Spring Data Type** |
| id | | number | int |
| courseName | | string | String |
| students | | StudentDTO[] | List<StudentDTO> |

|  |  |  |
| --- | --- | --- |
| **New** | **Class Name** | CourseVTORM |
| **Spring Path** | | model/course/rm |

|  |  |  |  |
| --- | --- | --- | --- |
| **New** | **Class Name** | Category | |
| **Angular Path** | | app/layout/shared/data/category.data.ts | |
| **Spring Path** | | model/lookup/Category | |
| **Fields** | | | |
| **Attribute** | | **Data Type** | **Description** |
| id | | number |  |
| labelEN | | string |  |

## Angular Component

|  |  |  |
| --- | --- | --- |
| **New** | **Class Name** | CreateCourseDetail |
| **Path** | | app/modules/course/components/create-course-details/create-course-details.component.[ts/html] |

|  |  |  |
| --- | --- | --- |
| **New** | **Class Name** | CourseService |
| **Path** | | app/modules/course/shared/course.service.ts |
| **Function** | | createNewCourse(data: CourseDTO) |

## Rest Services

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **New** | **Name** | | api/course/new | | | **Method** | POST |
| **Controller** | | | CourseRes | | | **Function** | createNewCourse(CourseDTO) |
| **REST Parameter** | | | | | | | |
| **Attribute** | | **Parameter Type** | | **Data Type** | **Description** | | |
| data | | Payload | | CourseDTO |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **New** | **Name** | | api/lookup/category | | | **Method** | GET |
| **Controller** | | | LookupRes | | | **Function** | getAllCategories() |
| **REST Parameter** | | | | | | | |
| **Attribute** | | **Parameter Type** | | **Data Type** | **Description** | | |
| N/A | | N/A | | N/A | N/A | | |

## Spring Service:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **New** | **Class Name** | CourseSer | **Function** | createNewCourse(CourseDTO data) |
| **Description** | | Add New Department | | |
| **Steps** | | | | |
| 1. Check if the user is Instructor 2. Validate Course Data 3. Insert Course Data in Database | | | | |

## Spring Repository:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **New** | **Class Name** | CourseRep | **Type** | Insert SQL |
| **Function** | | insertNewCourse(CourseDTO data) | | |
| **Query** | | | | |
| INSERT INTO course (cor\_name, duration, cor\_type\_id, cor\_level\_id) VALUES (?, ?, ?, ?) | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **New** | **Name** | LookupRep | **Type** | Select SQL |
| **Function** | | getAllCategories() | | |
| **Query** | | | | |
| SELECT id, label\_en from category | | | | |

## Database Script:

CREATE TABLE course (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

cor\_name VARCHAR(20) NOT NULL,

duration INT NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE,

instructor\_id INT NOT NULL,

category\_id INT NOT NULL,

type\_id INT NOT NULL,

level\_id INT NOT NULL,

description VARCHAR(200),

CONSTRAINT fk\_course\_cor\_category FOREIGN KEY (category\_id) REFERENCES cor\_category (id),

CONSTRAINT fk\_course\_cor\_type FOREIGN KEY (type\_id) REFERENCES cor\_type (id),

CONSTRAINT fk\_course\_cor\_level FOREIGN KEY (level\_id) REFERENCES cor\_level (id),

CONSTRAINT fk\_course\_auth\_user FOREIGN KEY (instructor\_id) REFERENCES auth\_user (id)

);

CREATE TABLE cor\_category (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

label\_en VARCHAR(45) NOT NULL

);

## ER Diagram: