

The Trinity Clique Team

GradeCenter Documentation

Written by Ivaylo Yotsov

The Trinity Clique Team

Ivaylo Yotsov (F099684) - Full Stack Developer, Team Lead, Scrum Master

Alexander Stoyanov (F099867) - Full Stack Developer

Ivan Kovachev (F110825) - Full stack Developer

How we work?

We use professional methodology for our organizational processes called SCRUM, its initiator is Ivaylo Yotsov. There are multiple types of meetings where we share our progress on the work, we discuss the next tasks and reflect on the previous sprint. The sprints are for 1 week only as we decided we won't need more.

What is the project about?

This is a project developed for New Bulgarian University. Its purpose is serving for managing multiple schools, their students and teachers. It was developed using .NET Core Restful API and React.

Controllers

AccountController

- `Register`: This method registers a new user in the database. It takes a username, email, and password as parameters.
- `Login`: This method logs in a user and returns a JWT token. It takes a username and password as parameters.
- `AddChild`: This method adds a child to a parent user. It takes a parent ID and the child's first and last names as parameters.
- `Update`: This method updates user details like password and phone number. It takes a user ID, new password, new role, activity status, and new phone number as parameters.
- `GetLoggedInUser`: This method retrieves the details of the currently logged-in user.
- `GetUserById`: This method retrieves a user by their ID.
- `GetAllUsers`: This method retrieves all users.

AttendancesController

- **Create:** This method creates a new attendance record in the database. It takes an `AttendanceRequestModel` object as a parameter, which contains the details of the attendance to be created.
- **GetAllAttendancesAsync:** This method retrieves all attendance records from the database. It does not require any parameters and returns a list of all attendances. This method can be accessed by users with the roles Admin, Principle, Student, and Teacher.
- **Update:** This method updates an existing attendance record in the database. It also takes an `AttendanceRequestModel` object as a parameter, which contains the updated details of the attendance.
- **Delete:** This method deletes an attendance record from the database. It takes an attendance ID as a parameter.

CurriculumController

- **GetDisciplineByTeacherId:** This method retrieves the discipline by a teacher's ID.
- **Create:** This method creates a new curriculum in the database. It takes a list of `DisciplineDto` objects as a parameter, which contains the details of the disciplines to be included in the curriculum.
- **Update:** This method updates an existing curriculum in the database. It also takes a list of `DisciplineDto` objects as a parameter, which contains the updated details of the disciplines in the curriculum.
- **Delete:** This method deletes disciplines from the curriculum. It takes a list of `DisciplineDto` objects as a parameter, which contains the details of the disciplines to be deleted from the curriculum.
- **GetLoggedUserClasses:** This method retrieves the disciplines associated with the currently logged-in user.
- **GetClassesForDay:** This method retrieves the disciplines for a given school class and day.
- **GetCurriculumForSchoolClass:** This method retrieves all disciplines for a given school class.
- **GetLoggedUserCurriculum:** This method retrieves the curriculum of the currently logged-in user.

GradesController

- **Create:** This method creates a new grade in the database. It takes a `GradeRequestModel` object as a parameter, which contains the details of the grade to be created.
- **GetAllGradesAsync:** This method retrieves all grades from the database. It does not require any parameters and returns a list of all grades. This method can be accessed by users with the roles Admin, Principle, Student, and Teacher.
- **Update:** This method updates an existing grade in the database. It also takes a `GradeRequestModel` object as a parameter, which contains the updated details of the grade.
- **Delete:** This method deletes a grade from the database. It takes a grade ID as a parameter.

SchoolController.cs

- **GetAllSchools:** Reads all the existing School entities in the database.
- **GetSchoolById:** Retrieves a school by its ID.
- **GetPeopleInSchool:** Retrieves the list of people in a school.
- **Create:** Creates a new school entity and stores it in the database.
- **Update:** Updates an existing school entity in the database.
- **AddPrincipal:** Adds a principal to a school.
- **AddTeachers:** Adds teachers to a school.
- **AddStudents:** Adds students to a school.
- **Delete:** Soft deletes an object of type School in the database.

SchoolClassController.cs

- `GetAllClasses`: This method retrieves all school classes from the database.
- `GetClassesInSchool`: This method retrieves all classes in a specific school. It takes a school ID as a parameter.
- `CreateClass`: This method creates a new class in the database. It takes a `SchoolClassCreateRequest` object as a parameter.
- `EnrollForClass`: This method enrolls a student in a class. It takes a `EnrollWithdrawRequestModel` object as a parameter.
- `WithdrawFromClass`: This method withdraws a student from a class. It takes a student ID as a parameter.

StatisticsController.cs

- `GetWeekly`: This method retrieves weekly statistics. It takes a `StatisticTypes` enumeration as a parameter.
- `GetMonthly`: This method retrieves monthly statistics. It takes a `StatisticTypes` enumeration as a parameter.
- `GetYearly`: This method retrieves yearly statistics. It takes a `StatisticTypes` enumeration as a parameter.
- `GetSchoolStatistics`: This method retrieves statistics for schools. It does not require any parameters.
- `GetClassStatistics`: This method retrieves statistics for classes. It does not require any parameters.
- `GetTeacherStatistics`: This method retrieves statistics for teachers. It does not require any parameters.
- `CreateGradesStatistics`: This method creates a new statistics record in the database. It takes optional `schoolId`, `schoolClassIdId`, `teacherId`, a `discipline`, and a `StatisticTypes` enumeration as parameters. The `StatisticTypes` enumeration indicates the type of statistics to create.

Services

- **SchoolClassService**

- `GetAllClasses()`: Returns a list of all classes, including related entities like the school, students, and head teacher.
- `GetSchoolClassByName(string schoolClassName)`: Returns a specific school class based on its name.
- `GetClassesInSchool(string schoolId)`: Returns a list of all classes in a specific school.
- `CreateClass(SchoolClass newSchoolClass)`: Creates a new class in the database.
- `EnrollForClass(string className, string studentId)`: Enrolls a student in a class.
- `WithdrawFromClass(string studentId)`: Withdraws a student from a class.
- `IsStudentInClass(SchoolClass schoolClass, AspNetUser user)`: Checks if a student is enrolled in a class.

- **SchoolService**

- `GetSchoolByName(string name)`: Returns a school based on its name.
- `GetSchoolById(string id)`: Returns a school based on its ID.
- `GetPeopleInSchool(string schoolId)`: Returns a list of all people in a specific school.
- `GetSchoolClassById(string id)`: Returns a specific class in a school based on its ID.
- `GetAllSchools()`: Returns a list of all schools.
- `Create(School newSchool)`: Creates a new school in the database.
- `Update(School? updatedSchool)`: Updates an existing school in the database.
- `Delete(string schoolName)`: Deletes a school from the database.

- **StatisticsService**

- `GetStatistics(StatisticTypes statisticType, DateTime start, DateTime end)`: Returns a list of statistics based on the type and date range.
- `GetMonthly(StatisticTypes statisticType)`: Returns a list of monthly statistics based on the type.
- `GetWeekly(StatisticTypes statisticType)`: Returns a list of weekly statistics based on the type.
- **StatisticsService (continued)**
- `GetYearly(StatisticTypes statisticTypes)`: Returns a list of yearly statistics based on the type.
- `GetSchoolStatistics()`: Returns a list of all school statistics.
- `GetClassStatistics()`: Returns a list of all class statistics.
- `GetTeacherStatistics()`: Returns a list of all teacher statistics.
- `CreateStatistic(string? schoolId, string? schoolClassId, string? teacherId, string? disciplineName, StatisticTypes statisticType)`: Creates a new statistic in the database.

- **AttendanceService**

- `Create`: This method creates a new attendance record in the database. It takes an `Attendance` object as a parameter.
- `Update`: This method updates an existing attendance record in the database. It takes an `Attendance` object as a parameter.
- `Delete`: This method deletes an attendance record from the database. It takes an attendance ID as a parameter.
- `GetAllAttendances`: This method retrieves all attendance records from the database.
- `GetAttendanceById`: This method retrieves an attendance record by its ID.

- **GradeService**

- **Create:** This method creates a new grade in the database. It takes a `Grade` object as a parameter.
- **Update:** This method updates an existing grade in the database. It takes a `Grade` object as a parameter.
- **Delete:** This method deletes a grade from the database. It takes a grade ID as a parameter.
- **GetAllGrades:** This method retrieves all grades from the database.
- **GetGradeById:** This method retrieves a grade by its ID.

- **AccountService**

- **Login:** This method logs in a user and returns a JWT token. It takes a username and password as parameters.
- **UpdateUser:** This method updates user details like password and phone number. It takes a user ID, new password, new role, activity status, and new phone number as parameters.
- **Register:** This method registers a new user in the database. It takes a username, email, and password as parameters.
- **Deactivate:** This method deactivates a user. It takes an `AspNetUser` object as a parameter.
- **GetUserById:** This method retrieves a user by their ID.
- **AddChild:** This method adds a child to a parent user. It takes a parent ID and the child's first and last names as parameters.
- **GetAllUsers:** This method retrieves all users.
- **GetUserByNames:** This method retrieves a user by their first and last names.

- **CurriculumService**

- **Create:** This method creates a new curriculum in the database. It takes a list of `Discipline` objects as a parameter.
- **Update:** This method updates an existing curriculum in the database. It takes a list of `Discipline` objects as a parameter.
- **Delete:** This method deletes disciplines from the curriculum. It takes a list of `Discipline` objects as a parameter.
- **GetClassesForDay:** This method retrieves the disciplines for a given school class and day.
- **GetLoggedInUserClasses:** This method retrieves the disciplines associated with the currently logged-in user.
- **GetCurriculumForSchoolClass:** This method retrieves all disciplines for a given school class.
- **GetDisciplineByTeacherId:** This method retrieves the discipline by a teacher's ID.

Models

AspNetUser

- **FirstName:** Represents the first name of the user. It is a required property.
- **LastName:** Represents the last name of the user. It is a required property.
- **SchoolId:** Represents the foreign key to the `School` entity. It is annotated with the `ForeignKey` attribute.
- **School:** Represents the navigation property to the associated `School` entity.
- **IsActive:** Represents the status of the user. It is a nullable boolean property.
- **UserRole:** Represents the role of the user. It is an enumeration of `UserRoles`.
- **SchoolClass:** Represents the associated school class for the user.
- **ChildrenRelations:** Represents a collection of child relations for the user. It is a collection of `UserRelation` entities.
- **ParentRelations:** Represents a collection of parent relations for the user. It is a collection of `UserRelation` entities.
- **Statistics:** Represents a collection of statistics associated with the user. It is a collection of `Statistic` entities.
- **Attendances:** Represents a collection of attendances associated with the user. It is a collection of `Attendance` entities.
- **Grades:** Represents a collection of grades associated with the user. It is a collection of `Grade` entities.

Attendance

- **Id:** Represents the unique identifier for the attendance.
- **Date:** Represents the date of the attendance.
- **Discipline:** Represents the associated discipline for the attendance.
- **IsActive:** Represents the status of the attendance.
- **Student:** Represents the associated student for the attendance.
- **HasAttended:** Represents whether the student has attended the discipline or not.

Discipline

- **Id:** Represents the unique identifier for the discipline.
- **Name:** Represents the name of the discipline.
- **OccuranceDay:** Represents the day of the week when the discipline occurs.
- **OccuranceTime:** Represents the time of occurrence for the discipline.
- **SchoolClassId:** Represents the foreign key to the associated school class.
- **SchoolClass:** Represents the associated school class for the discipline.
- **TeacherId:** Represents the foreign key to the associated teacher.
- **Teacher:** Represents the associated teacher for the discipline.
- **IsActive:** Represents the status of the discipline.
- **Attendances:** Represents a collection of attendances associated with the discipline.
- **Grades:** Represents a collection of grades associated with the discipline.
- **Statistics:** Represents a collection of statistics associated with the discipline.

Grade

- **Id:** Represents the unique identifier for the grade.
- **Rate:** Represents the rate of the grade.
- **IsActive:** Represents the status of the grade.
- **Discipline:** Represents the associated discipline for the grade.
- **Student:** Represents the associated student for the grade.

School

- **Id:** Represents the unique identifier for the school.
- **Name:** Represents the name of the school.
- **Address:** Represents the address of the school.
- **IsActive:** Represents the status of the school.
- **People:** Represents a collection of people associated with the school.
- **SchoolClasses:** Represents a collection of school classes associated with the school.
- **Statistics:** Represents a collection of statistics associated with the school.

SchoolClass

- **Id**: Represents the unique identifier for the school class.
- **Year**: Represents the year of the school class.
- **Department**: Represents the department of the school class.
- **HeadTeacher**: Represents the head teacher of the school class.
- **School**: Represents the associated school for the school class.
- **Students**: Represents a collection of students in the school class.
- **Curriculum**: Represents a collection of disciplines in the school class curriculum.
- **Statistics**: Represents a collection of statistics associated with the school class.

Statistic

- **Id**: Represents the unique identifier for the statistic.
- **AverageRate**: Represents the average rate of the statistic (grade or attendance).
- **ComparedToLastWeek**: Represents the comparison of this week's average rate with the last week's average rate in percentage change.
- **ComparedToLastMonth**: Represents the comparison of this week's average rate with the last month's average rate in percentage change.
- **ComparedToLastYear**: Represents the comparison of this week's average rate with the last year's average rate in percentage change.
- **SchoolClass**: Represents the associated school class for the statistic.
- **School**: Represents the associated school for the statistic.
- **Teacher**: Represents the associated teacher for the statistic.
- **DisciplineName**: Represents the name of the discipline for the statistic.
- **CreatedOn**: Represents the creation date of the statistic.
- **StatisticType**: Represents the type of the statistic (grades or attendance).