

# API Documentation

Endpoints:

Student					
Name	Method	URL	Parameter	Response	Description
Get student by id (done)	Get	student/{id}		Get a <b>Student</b> model with <b>200</b> response status as result.	
Update student (done)	Put	student/{id}	'givenName' => 'string max:255' 'surName' => 'string max:255', 'initials' => 'string max:255', 'displayName' => 'string max:255', => 'string max:255'  <b>Notes:</b> none of the parameters are required.	When a student is updated successfully a response status code <b>200</b> is returned.	
Delete student (done)	Delete	student/{id}		When a student is deleted successfully a response status code <b>202</b> is returned.	
Get all students (done)	Get	students/all		Get an array of <b>Student</b> models with <b>200</b> response status as result.	
Get all skills of student (done)	Get	student/{id}/skill/all		Get an array of <b>Skill</b> models with <b>200</b> response status as result.	
Add Skill To Student (done)	Post	student/{id}/skill	<b>studentId:int, skill:String or enum,</b>  'skill' => 'required string max:255' 'type' => 'required string max:255 in:soft skill,hard skill', 'level' => 'required integer between:1,5'  <b>Note:</b> Skill.type can have only two values “soft skill” or “hard skill”.  <b>Note:</b> Level can be a number between 1 and 5		
Update Student Skill (done)	Put	student/{id}/skill	'skill' => 'string max:255', 'type' => 'string max:255 in:soft skill,hard skill'  <b>Notes:</b> none of the parameters are required.		
Delete Student Skill (done)	Delete	student/{studentId}/skill/{skillId}			
Project					
Name	Method	URL	Parameter		
Create project	Post	projects/			Create project first without specifying any spot
Update project	Put	projects/{id}		Receive project model	
Delete project	Delete	projects/{id}			
Get all projects	Get	projects/all			
Get project by id	Get	projects/{id}			

Get all project's join requests	Get	projects/{projectId}/requests			
Create Join Request	Post	projects/requests	studentId, projectId		
Accept Join Request	Delete	project/requests/accept	requestId		Delete record in request table and add student to a spot in a project
Create Spot	Post	projects/spot	projectId	All fields in Spot model + array of skills related to a spot(an int array)	Create spot in the project, this is done after a project is created
Add student to project	Put	projects/addstudent	projectId,studentId		Probably remove this
Reject student request	Delete	projects/requests/reject	requestId		
Delete student from project	Put	project/{projectId}/{studentId}/student/	projectId, studentId		
Get Role	Get	project/role/all			
Delete spot	Delete	project/spot	spotid		
Search					
Search Projects	Get				
Typeahead function	Get				
Filtering	Get				

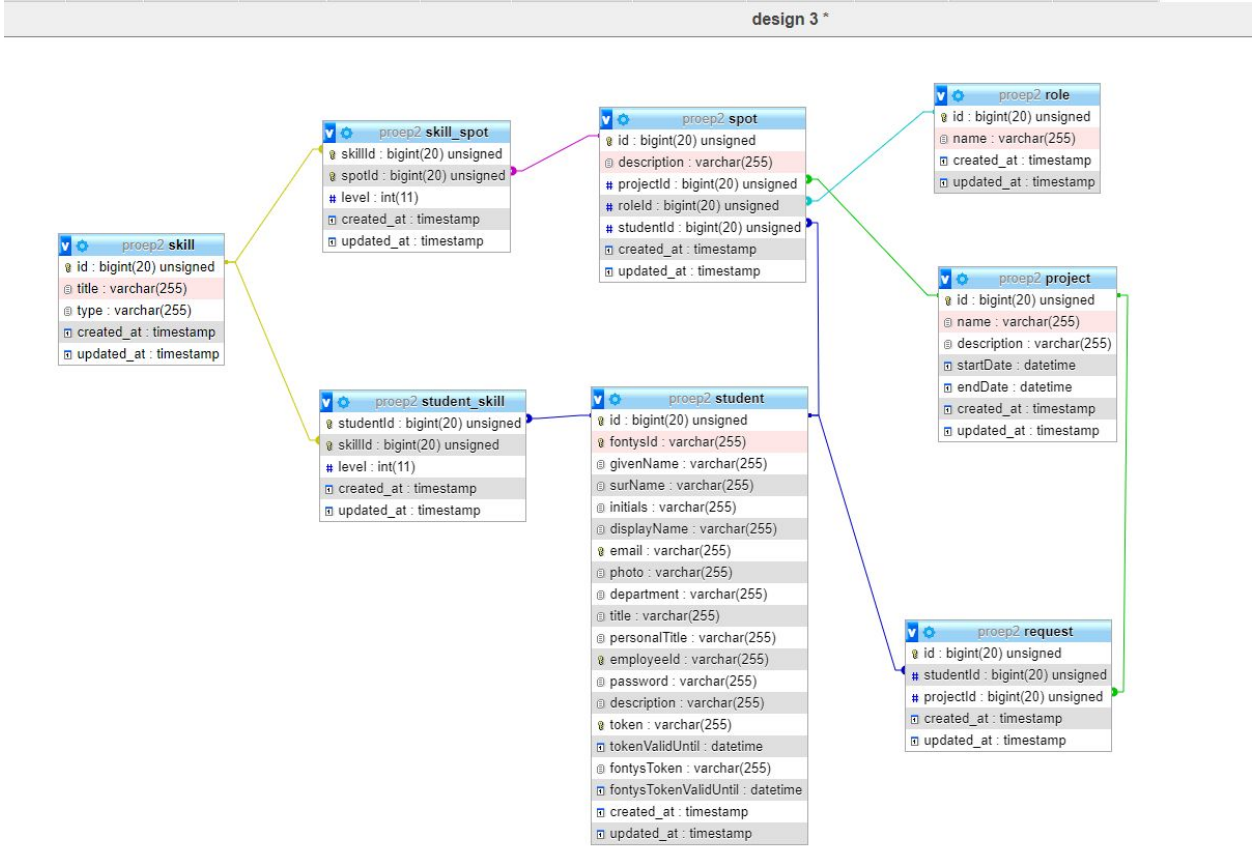
Skill			
Name	Method	URL	Parameter
Get all the skills (done)	Get	student/{id}	
Get a skill by Id (done)	Get	student/{id}	
Add skill (done)	Post	student/{id}	'skill' => 'required string max:255', 'type' => 'required string max:255 in:soft skill,hard skill'
Update a skill (done)	Put	students/all	
Delete a skill (done)	Delete	student/{id}/skill/all	

Responses

```
Student
{
  "Id": "String",
  "fontysId": "String",
  "givenName": "String",
  "surName": "String",
  "initials": "String",
  "displayName": "String",
  "email": "String",
  "photo": "String",
  "department": "String",
  "title": "String",
  "personalTitle": "String",
```

```
"employeeId": "String",  
"password": "String",  
"description": "String",  
"token": "String",  
"tokenValidUntil": "String",  
"fontysToken": "String",  
"fontysTokenValidUntil": "DateTime",  
"created_at": "DateTime",  
"updated_at": "DateTime",  
}
```

Database design



studentName

Get student

```
[
  {
    Student entity:
    List<project>:

  }
}
```

- The level of a skill is deleted.
- Upload profile picture (update)
- Endorsement. Only the group members can endorse a skill.