

Documentação de Testes – Projects and Teams API

Métodos de execução e validação

Todos os testes foram realizados através da interface gráfica do Swagger. Para definição dos resultados esperados, foi levada em consideração a documentação previamente disponibilizada juntamente com os refinamentos dos documentos produzidos pelo grupo.

Resultados obtidos

Foram realizados um total de 32 testes manuais considerando as rotas de custos, cargos, skills e colaboradores. 25% dos testes não ocorreram devido à falta da implementação no banco de dados, o que implica em um quarto das funcionalidades.

Custos

Testes executados no projeto:

- Criar novo Custo;
- Visualização de lista de Custos;
- Busca de Custo por descrição;
- Paginação de listas;
- Busca de Custo por ID;
- Remoção de Custos;
- Edição de Custos;
- Busca de Custos por período;
- Busca de Custos por ID de Projeto;
- Busca de Custos por ID de Cargo;

Nome	Tipo	Rota	IN	OUT	Esperado	Teste	Comentário
Criar novo Custo	POST	/costs/	{ "id": "string", "valor": 0, "dataInicio": "string", "dataFim": "string", "descricao": "string" }	{ "id": "string", "valor": 0, "dataInicio": "string", "dataFim": "string", "descricao": "string" }	Criação de um custo com os dados de entrada e adição no repositório.	Pass	Com a criação do banco de dados, o id deve ser gerado automaticamente como uma chave primária(único). As datas devem ser preenchidas no formato YYYY-MM-DD
Buscar Custos	GET	/costs/		json com lista de custos	Retorna uma lista com os custos registrados no repositório.	Pass	Aceita parâmetros adicionais tipo query
Buscar Custos(Paginado)	GET	/costs/?qtdItens=2&pagina=2		json contendo lista de até 2 custos	Retorna uma lista contendo a segunda página dos custos registrados. Tamanho de página = 2.	Pass	Retorna uma lista vazia, caso não tenha a devida correspondência
Buscar Custos por descricao	GET	/costs/?descricao=palavra		json contendo lista de custos	Retorna uma lista contendo os custos cuja descrição possua a string "palavra" em algum trecho.	Pass	Também é possível aplicar paginação junto com a busca por descrição.
Buscar Custos por ID de Cargo	GET	/costs/cargo/{id}		json contendo lista de custos	Retorna uma lista de custos correspondentes ao cargo especificado.	Fail	É necessário a implementação do banco de dados para cruzar esses dados.
Buscar Custos por ID de Projeto	GET	/costs/projeto/{id}		json contendo lista de custos	Retorna uma lista de custos correspondentes ao projeto especificado.	Fail	É necessário a implementação do banco de dados para cruzar esses dados.

Editar Custos	PUT	/costs/{id}	{ "valor": 5, "dataInicio": "string", "dataFim": "string", "descricao": "string" }	{ "id": "id", "valor": 5, "dataInicio": "string", "dataFim": "string", "descricao": "string", }	Edita os dados do custo identificado pelo id, substituindo-os pelos dados contidos no json de entrada	Pass	
Buscar Custo por ID	GET	/costs/{id}		json com custo	Retorna um único custo identificado pelo id	Pass	
Remover Custo	DELETE	/costs/{id}		{ "id": "1", "valor": 5, "dataInicio": "string", "dataFim": "string", "descricao": "string", "deletedBy": "", "deletedDate": "2021-12-09 02:05:03.597055" }	Marca custo como removido ao preencher o item deletedDate no custo específico	Pass	O campo deletedBy não está sendo preenchido pela falta de login com um banco de dados, para verificar o usuário logado.
Buscar Custo por período de tempo	GET	/costs/{inicio}/{fim}		json contendo lista de custos do período informado	Retorna lista de custos que intersectam com o período delimitado pelos parâmetros início e fim.	Pass	As datas devem ser preenchidas no formato YYYY-MM-DD É possível utilizar parâmetros de query de paginação neste item(qtdItens e pagina).

Skills

Testes executados no projeto:

- Criar nova Skill;
- Buscar Skills;
- Editar dados de uma Skill;
- Buscar Skills por ID;
- Deletar uma Skill;
- Buscar skill por ID de colaborador.

Nome	Tipo	Rota	IN	OUT	Esperado	Teste	Comentário
Criar nova Skill	POST	/skills/	{ "id": "string", "descricao": "string" }	{ "id": "string", "descricao": "string" }	Criação de uma nova skill com os dados de entrada e adição no repositório.	Pass	Com a criação do banco de dados, o id deve ser gerado automaticamente como uma chave primária(único).
Buscar Skills	GET	/skills/		json com lista de skills	Retorna uma lista com as skills registradas no repositório.	Pass	Aceita parametros adicionais tipo query
Buscar Skill por ID de colaborador	GET	/skills/colaborador/{id}		json contendo lista de skills	Retorna uma lista de skills correspondentes ao colaborador especificado.	Fail	É necessário a implementação do banco de dados para cruzar esses dados.
Editar Skills	PUT	/skills/{id}	{ "descricao": "string" }	{ "id": "id", "descricao": "string" }	Edita os dados de uma skill identificado pelo id, substituindo-os pelos dados contidos no json de entrada	Pass	

Buscar Skill por ID	GET	/skills/{id}		json com custo	Retorna uma única skill identificado pelo id	Pass	
Remover Skill	DELETE	/skills/{id}		{ "id": "1", "descricao": "Skill de id 1" }	Apaga a skill da lista de skills cadastrados. Mensagem de feedback é exibida ao final da operação	Pass	

Colaborador

Testes executados no projeto:

- Criar novo colaborador;
- Visualizar lista de colaborador;
- Editar dados de um colaborador;
- Visualizar dados de um colaborador;
- Deletar um colaborador;
- Buscar colaborador através de cargo e skill.

Nome	Tipo	Rota	IN	OUT	Esperado	Teste	Comentário
Criar novo	POST	/colaborador	{ id: string funcao: string descricao: string nome: string }	{ id: string funcao: string descricao: string nome: string }	Novo cargo é adicionado à lista de cargos existentes	Pass	
Visualizar lista	GET	/colaborador		json com lista de cargos	Visualizar uma lista com somente o título dos cargos	Pass	

Editar dados	PUT	/colaborador	{ id: string funcao: string descricao: string nome: string }	{ id: string funcao: string descricao: string nome: string }	Mesmo que o ID seja editado, o ID passado como parâmetro permanece inalterado. Campos de nome, descrição e função foram atualizados	Pass	
Visualizar dados	GET	/colaborador/id		{ id: string funcao: string descricao: string nome: string }	Retorna o ID, nome, descrição e função de um cargo	Pass	
Deletar	DELETE	/colaborador/id			Remove o cargo da lista de cargos cadastrados	Pass	
Buscar: por setor	GET	/colaborador/cargo			Retorna uma lista de colaborador que estejam presentes no setor passada como parâmetro	Fail	Como não houve implementação do BD, não foi possível prosseguir com a funcionalidade
Buscar: por equipe	GET	/colaborador/skill/id			Retorna uma lista de colaborador presentes na equipe passada como parâmetro	Fail	Como não houve implementação do BD, não foi possível prosseguir com a funcionalidade

Cargos

Testes executados no projeto:

- Criar novo cargo;
- Visualizar lista de cargos;
- Editar dados de um cargo;
- Visualizar dados de um cargo;
- Deletar um cargo;

- Buscar cargo através de nome, equipe e skill.

Resultado das execuções

Nome	Tipo	Rota	IN	OUT	Esperado	Teste	Comentário
Criar novo	POST	/cargos	{ id: string funcao: string descricao: string nome: string }	{ id: string funcao: string descricao: string nome: string }	Novo cargo é adicionado à lista de cargos existentes	Pass	
Visualizar lista	GET	/cargos		json com lista de cargos	Visualizar uma lista com somente o título dos cargos	Pass	
Editar dados	PUT	/cargos	{ id: string funcao: string descricao: string nome: string }	{ id: string funcao: string descricao: string nome: string }	Mesmo que o ID seja editado, o ID passado como parâmetro permanece inalterado. Campos de nome, descrição e função foram atualizados	Pass	
Visualizar dados	GET	/cargos/id		{ id: string funcao: string descricao: string nome: string }	Retorna o ID, nome, descrição e função de um cargo	Pass	
Deletar	DELETE	/cargos/id			Remove o cargo da lista de cargos cadastrados	Pass	

Buscar: por nome	GET	/cargos/findByName	{ nome: string }	{ id: string funcao: string descricao: string nome: string }	Retorna uma lista com os cargos cadastrados que possuam o nome igual ao parâmetro passado	Pass	
Buscar: por setor	GET	/cargos/findBySect			Retorna uma lista de cargos que estejam presentes no setor passada como parâmetro	Fail	Como não houve implementação do BD, não foi possível prosseguir com a funcionalidade
Buscar: por equipe	GET	/cargos/squad/id			Retorna uma lista de cargos presentes na equipe passada como parâmetro	Fail	Como não houve implementação do BD, não foi possível prosseguir com a funcionalidade
Buscar: por skill	GET	/cargos/skill/id			Retorna uma lista de cargos que possuem a skill passada como parâmetro	Fail	Como não houve implementação do BD, não foi possível prosseguir com a funcionalidade