

Curso .NET

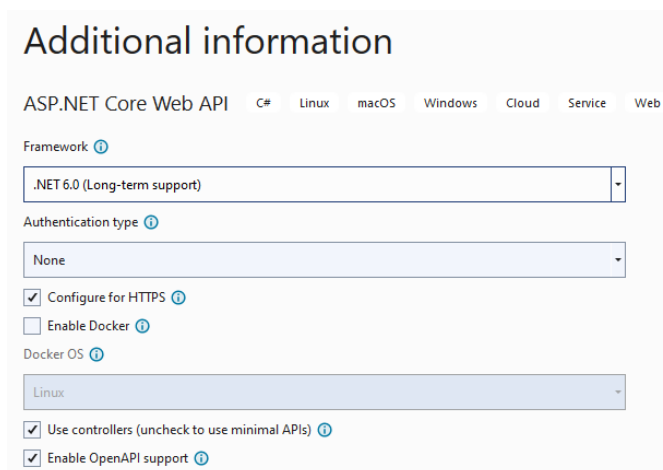
Unidade Curricular: Laboratório Web

Docente: David Jardim

FICHA DE TRABALHO 10

Exercícios:

1. Projeto (*API with controllers*)
 - a. Crie um projeto do tipo *Web API* denominado por *Ficha10* com as seguintes definições:



Additional information

ASP.NET Core Web API C# Linux macOS Windows Cloud Service Web

Framework ⓘ

.NET 6.0 (Long-term support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ

Linux

☒ Use controllers (unchecked to use minimal APIs) ⓘ

☒ Enable OpenAPI support ⓘ

2. JSON e Modelos
 - a. Adicione os ficheiros JSON facultados ao projeto (*employees.json* e *characters.json*)
 - b. Crie uma pasta denominada por *Models*
 - c. Crie uma classe *Employee* para representar um funcionário e adicione as propriedades referidas na alínea anterior (tenha em atenção os tipos) e respeite os nomes do ficheiro JSON
 - d. Crie uma classe *Employees* com uma única propriedade do tipo *List* para armazenar instâncias do tipo *Employee*
 - e. Crie uma classe *Character* para representar uma personagem e adicione as propriedades referidas na alínea anterior (tenha em atenção os tipos) e respeite os nomes do ficheiro JSON (1 valor)
 - f. Crie uma classe *Characters* com uma única propriedade do tipo *List* para armazenar instâncias do tipo *Character*
 - g. Crie uma classe *JsonLoader* que será responsável por ler todos os ficheiros JSON
 - i. Implemente uma função estática para ler o ficheiro **employees.json** desserializar o objeto, criar uma nova instância do tipo *Employees* e devolvê-la
 - ii. Implemente uma função para ler o ficheiro **characters.json**, desserializar o objeto, criar uma nova instância do tipo *Characters* e devolvê-la

3. Controllers

- a. Tendo em conta a Tabela 1, na pasta *Controllers* implemente um *controller* `EmployeesController` com os seguintes endpoints tendo em conta as validações necessárias e respostas adequadas:
 - i. Listar todos os funcionários existentes no ficheiro e devolver os mesmos na resposta
 - ii. Adicionar um novo funcionário, o ID deve ser gerado automaticamente tendo em conta o número de funcionários existentes. O ID do novo funcionário deve ser devolvido na resposta
 - iii. Apagar um funcionário pelo seu ID. O ID do funcionário removido deve ser devolvido na resposta
 - iv. Selecionar apenas um funcionário pelo seu ID e devolver esse mesmo funcionário na resposta
 - v. Alterar os detalhes de um determinado funcionário pelo seu ID e devolver essa mesma funcionário atualizada na resposta
 - vi. Listar todos os funcionários por região
 - vii. Efetuar download da lista atual de funcionários como um ficheiro .json
- b. Tendo em conta a Tabela 2, na pasta *Controllers* implemente um *controller* `CharactersController` com os seguintes endpoints tendo em conta as validações necessárias e respostas adequadas:
 - i. Listar todas as personagens existentes no ficheiro e devolver as mesmas na resposta
 - ii. Adicionar uma nova personagem, o ID deve ser gerado automaticamente tendo em conta o número de personagens existentes. O ID da nova personagem deve ser devolvido na resposta
 - iii. Apagar uma personagem pelo seu ID. O ID da personagem removida deve ser devolvido na resposta
 - iv. Selecionar apenas uma personagem pelo seu ID e devolver essa mesma personagem na resposta
 - v. Alterar os detalhes de uma determinada personagem pelo seu ID e devolver essa mesma personagem atualizada na resposta
 - vi. Listar as personagens por género
 - vii. Listar todas as personagens que são Jedi
 - viii. Efetuar download da lista atual de personagens como um ficheiro .json

URI	Método HTTP	Body do POST	Resultado
/employees	GET	empty	List all employees
/employees	POST	JSON String	Add details of a new employee
/employees/{id}	DELETE	empty	Delete an existing employee
/employees/{id}	GET	empty	Show details of an employee
/employees/{id}	PUT	JSON String	Update details of an employee
/employees/{region}	GET	empty	Show employees from a region
/employees/download	GET	empty	Download list of employees

Tabela 1 - Métodos a implementar

URI	Método HTTP	Body do POST	Resultado
/characters	GET	empty	List all characters
/characters	POST	JSON String	Add details of a new character
/characters/{id}	DELETE	empty	Delete an existing character
/characters/{id}	GET	empty	Show details of a character
/characters/{id}	PUT	JSON String	Update details of a character
/characters/{gender}	GET	empty	Show characters of a gender
/characters/jedi	GET	Empty	List all characters that are Jedi
/characters/download	GET	empty	Download list of characters

Tabela 2 . Métodos a implementar