

Enunciado: API de Controle de Disciplinas e Tarefas de Aluno

Você deve criar uma API em Django para ajudar os alunos a gerenciarem suas disciplinas e tarefas. A API deve ter os seguintes modelos:

Modelo Aluno:

Campos: nome, email

Modelo Disciplina:

Campos: nome, descricao

Modelo Tarefa:

Campos: titulo, descricao, data_entrega, concluida

Relacionamento Many-to-One com Aluno: Uma tarefa pertence a um aluno.

Relacionamento Many-to-Many com Disciplina: Uma tarefa pode estar associada a uma ou mais disciplinas.

A API deve incluir as seguintes funcionalidades:

1. Listagem de Alunos:
 - Endpoint: /api/alunos/
 - Método: GET
 - Descrição: Retorna a lista de todos os alunos.
2. Criação de um Aluno:
 - Endpoint: /api/alunos/
 - Método: POST
 - Descrição: Permite a criação de um novo aluno.
3. Detalhes do Aluno:
 - Endpoint: /api/alunos/<id>/
 - Método: GET
 - Descrição: Retorna detalhes de um aluno específico com base no id.
4. Atualização de um Aluno:
 - Endpoint: /api/alunos/<id>/
 - Método: PUT
 - Descrição: Permite a atualização dos detalhes de um aluno específico com base no id.
5. Exclusão de um Aluno:
 - Endpoint: /api/alunos/<id>/
 - Método: DELETE
 - Descrição: Permite a exclusão de um aluno específico com base no id. Além disso, todas as tarefas associadas a esse aluno devem ser excluídas ou desassociadas.
6. Listagem de Disciplinas:
 - Endpoint: /api/disciplinas/
 - Método: GET
 - Descrição: Retorna a lista de todas as disciplinas.

7. Criação de uma Disciplina:
 - Endpoint: /api/disciplinas/
 - Método: POST
 - Descrição: Permite a criação de uma nova disciplina.
8. Detalhes da Disciplina:
 - Endpoint: /api/disciplinas/<id>/
 - Método: GET
 - Descrição: Retorna detalhes de uma disciplina específica com base no id.
9. Atualização de uma Disciplina:
 - Endpoint: /api/disciplinas/<id>/
 - Método: PUT
 - Descrição: Permite a atualização dos detalhes de uma disciplina específica com base no id.
10. Exclusão de uma Disciplina:
 - Endpoint: /api/disciplinas/<id>/
 - Método: DELETE
 - Descrição: Permite a exclusão de uma disciplina específica com base no id. Certifique-se de que as tarefas associadas a essa disciplina sejam desassociadas.
11. Listagem de Tarefas:
 - Endpoint: /api/tarefas/
 - Método: GET
 - Descrição: Retorna a lista de todas as tarefas.
12. Criação de uma Tarefa:
 - Endpoint: /api/tarefas/
 - Método: POST
 - Descrição: Cria uma nova tarefa associada a um aluno e, a uma ou mais disciplinas.
13. Detalhes de uma Tarefa:
 - Endpoint: /api/tarefas/<id>/
 - Método: GET
 - Descrição: Retorna detalhes de uma tarefa específica com base no id.
14. Atualização de uma Tarefa:
 - Endpoint: /api/tarefas/<id>/
 - Método: PUT
 - Descrição: Atualiza os detalhes de uma tarefa específica com base no id.
15. Exclusão de uma Tarefa:
 - Endpoint: /api/tarefas/<id>/
 - Método: DELETE
 - Descrição: Exclui uma tarefa específica com base no id.
16. Listagem de Tarefas de um Aluno:
 - Endpoint: /api/alunos/<aluno_id>/tarefas/
 - Método: GET
 - Descrição: Retorna a lista de todas as tarefas associadas a um aluno específico.

Certifique-se de implementar as views baseadas em APIView para cada endpoint, e de que todas as operações de criação, atualização e exclusão de alunos e disciplinas sejam tratadas de forma segura, levando em consideração as relações com as tarefas e aplicando as devidas validações.

Além disso, documente sua API para que outros desenvolvedores possam facilmente entender como usá-la, incluindo exemplos de solicitações e respostas, de modo que os desenvolvedores possam compreender e usar todas as operações disponíveis na API de forma eficaz.

Critérios de Avaliação:

1. Funcionalidade Geral da API:
 - A API está funcional e responde corretamente a todas as solicitações HTTP (GET, POST, PUT, DELETE) para todas as entidades (alunos, disciplinas e tarefas).
2. Validação de Dados:
 - A API realiza validações adequadas dos dados de entrada, incluindo a verificação de datas, formatos de e-mail, campos obrigatórios e outros requisitos específicos para cada entidade.
3. Relacionamentos de Banco de Dados:
 - Os relacionamentos many-to-one (alunos-tarefas) e many-to-many (disciplinas-tarefas) são configurados corretamente e funcionam de acordo com os requisitos.
4. Documentação da API:
 - A documentação da API está completa e fornece informações detalhadas sobre como usar cada endpoint, incluindo exemplos de solicitações e respostas.
5. Manipulação de Erros:
 - A API lida adequadamente com erros, fornecendo respostas de erro apropriadas com códigos de status HTTP significativos e mensagens de erro descritivas.
6. Operações de Alunos:
 - Todas as operações relacionadas aos alunos (listagem, detalhes, criação, atualização, exclusão) funcionam corretamente e de acordo com as especificações.
7. Operações de Disciplinas:
 - Todas as operações relacionadas às disciplinas (listagem, detalhes, criação, atualização, exclusão) funcionam corretamente e de acordo com as especificações.
8. Operações de Tarefas:
 - Todas as operações relacionadas às tarefas (listagem, detalhes, criação, atualização, exclusão) funcionam corretamente e de acordo com as especificações.
9. Consistência de Nomenclatura:
 - Os nomes de endpoints, campos de modelos e respostas da API seguem uma nomenclatura consistente e fácil de entender, seguindo as convenções recomendadas.

Nível da Tarefa: 2