

Teste de codificação de desenvolvedor

Certifique-se de ler **este** documento com atenção e siga as diretrizes.

Descrição do Problema

ZSSN (Rede Social de Sobrevivência Zumbi). O mundo como o conheceu caiu em um cenário apocalíptico. Um vírus produzido em laboratório está transformando seres humanos e animais em zumbis, famintos por carne fresca.

Você, como membro da resistência aos zumbis (e o último sobrevivente que sabe codificar), foi designado para desenvolver um sistema para compartilhar recursos entre humanos não infectados.

Exigências

Use os seguintes Linguagens/ estruturas: *Python e os Frameworks Django e Django Rest. Banco de Dados Postgres. Ferramenta de Controle de Versões Git. HTML, CSS, Javascript.*

Você desenvolverá uma **API REST** (sim, nós nos preocupamos com o design da arquitetura, mesmo no meio de um apocalipse zumbi!), que armazenará informações sobre os sobreviventes, bem como os recursos que eles possuem.

Para fazer isso, a API deve atender aos seguintes casos de uso:

- **Adicionar sobreviventes ao banco de dados**

Um sobrevivente deve ter um *nome*, *idade*, *sexo* e *último local* (*latitude*, *longitude*).

Um sobrevivente também possui um inventário de recursos de sua própria propriedade (que você precisa declarar quando do registro do sobrevivente).

- **Atualizar local do sobrevivente**

Um sobrevivente deve ter a capacidade de atualizar seu último local, armazenando o novo par de latitude / longitude na base (não é necessário rastrear locais, basta substituir o anterior).

- **Sinalizar sobrevivente como infectado**

Em uma situação caótica como essa, é inevitável que um sobrevivente seja contaminado pelo vírus. Quando isso acontece, precisamos sinalizar o sobrevivente como infectado.

Um sobrevivente infectado não pode negociar com outras pessoas, não pode acessar / manipular seu inventário, nem ser listado nos relatórios (as pessoas

infectadas estão meio mortas de qualquer maneira, veja o item nos relatórios abaixo).

Um sobrevivente é marcado como infectado quando pelo menos outros três sobreviventes relatam sua contaminação.

Quando um sobrevivente é infectado, seus itens de inventário ficam inacessíveis (eles não podem negociar com outros).

- **Os sobreviventes não podem adicionar / remover itens do inventário**

Seus pertences devem ser declarados quando forem registrados pela primeira vez no sistema. Depois disso, eles só podem alterar seu inventário por meio de negociação com outros sobreviventes.

Os itens permitidos no inventário são descritos acima no primeiro recurso.

- **Itens comerciais:**

Os sobreviventes podem trocar itens entre si.

Para fazer isso, eles devem respeitar a tabela de preços abaixo, onde o valor de um item é descrito em termos de pontos.

Ambos os lados do comércio devem oferecer a mesma quantidade de pontos. Por exemplo, 1 água e 1 medicamento ($1 \times 4 + 1 \times 2$) valem 6 munições (6×1) ou 2 itens alimentares (2×3).

As operações em si não precisam ser armazenadas, mas os itens devem ser transferidos de um sobrevivente para o outro.

Item	Pontos
1 Água	4 pontos
1 Alimentação	3 pontos
1 Medicação	2 pontos

1 Munição	1 ponto

- **Relatórios**

A API deve oferecer os seguintes relatórios:

- i. Porcentagem de sobreviventes infectados.
- ii. Porcentagem de sobreviventes não infectados.
- iii. Quantidade média de cada tipo de recurso por sobrevivente (por exemplo, 5 águas por sobrevivente)
- iv. Pontos perdidos por causa do sobrevivente infectado.

Notas

1. Nenhuma autenticação é necessária (é um apocalipse zumbi, ninguém tentará invadir um sistema enquanto estiver fugindo de uma horda de zumbis);
2. Ainda nos preocupamos com técnicas adequadas de programação e arquitetura; você deve mostrar ser digno de superar o apocalipse zumbi através da força de suas habilidades;
3. Não se esqueça de fazer pelo menos uma documentação mínima dos pontos de extremidade da API e como usá-los;
4. Você deve escrever pelo menos alguns testes automatizados;
5. Na descrição do problema acima, você pode fazer uma solução muito simples ou adicionar recursos opcionais que não estão descritos. Use seu tempo com sabedoria; a solução ótima absoluta pode levar muito tempo para ser eficaz no apocalipse, portanto, você deve encontrar a melhor solução possível que se mantenha no menor espaço de tempo e ainda consiga mostrar suas habilidades para provar seu valor.
6. Escreva mensagens de confirmação concisas e claras, dividindo suas alterações em pequenos pedaços.
7. Realizar deploy da aplicação na plataforma de sua preferência.
8. Disponibilizar o código através do GitHub.
9. O teste deve ser entregue até o dia 12/08/2022.