Prova Técnica – C# .Net – Nível Sênior

Instruções

- Para essa tarefa você é livre para escolher qualquer tecnologia .Net que for mais favorável para cada aspecto da implementação. O código deverá ser escrito em C#. Se você decidir usar um framework que talvez não seja o melhor candidato mas você o escolheu porque estás mais familiarizado com ele, não tem problema, desde que você indique o porquê da escolha (i.e. Eu escolhi o framework X, mas o framework Y seria o ideal para acesso ao banco de dados porque a) b) c) ...). Não é esperado que você conheça novos frameworks para essa tarefa desde que você consiga explicar o porquê das suas escolhas claramente.
- É esperado também que a aplicação seja entregue funcional, então se houver qualquer condição especial para que ela seja configurada por favor providencie-as junto com o código.
- É esperado que o código seja de qualidade então não o escreva como “é só uma demo” mas como você o faria para um produto real. Você deve considerar a tarefa completa quando você considera o seu código “pronto para produção/implantação”.
- Conhecimento mínimo desejável: Injeção de dependência, DDD e TDD;
- Se você tiver qualquer dúvida por favor entre em contato conosco.

Requisito

Você está trabalhando para uma empresa de jogos online que opera vários servidores de jogos. Cada jogo resulta em ganho ou perda de pontos para o jogador.

Dados são mantidos em memória por cada servidor e periodicamente esses dados são persistidos. Sua tarefa é implementar um serviço que exponha 2 endpoints:

Endpoint 1:

Permite que os servidores persistam os dados do resultado de um jogo:
GameResult:

- playerId (long) – ID do jogador
- gameId (long) – ID do jogo
- win (long) – o número de pontos ganhos (positivos ou negativos)
- timestamp (date time) – data de quando o jogo foi realizado(UTC)

Como resultado da chamada a esse endpoint o balanço dos pontos do jogador devem ser, em algum momento, atualizados no banco de dados. Se um jogador não tem um registro do balanço dos seus pontos no banco de dados, ele deverá ser criado. NOTA: Um grupo de dados pode conter diversos registros de um único jogador (i.e. o jogador participou em vários jogos). Existem múltiplos servidores de jogos, realizando partidas simultâneas de jogos diferentes, então o serviço irá receber várias requisições concorrentes, que podem conter resultados do mesmo jogador. Inicialmente este serviço irá rodar como um piloto em um único servidor. Portanto, dados perdidos devido ao mal funcionamento do servidor ou do serviço não é considerado crítico, mas não deveria ocorrer dentro de circunstâncias normais.

Endpoint 2

Esse endpoint permite que os web sites onde o jogador inicia os jogos mostre um placar da classificação dos 100 melhores jogadores. Os 100 melhores jogadores são ordenados pelo balanço de pontos que eles possuem em ordem descendente. Ele retornará os seguintes dados:
Leaderboard:

- playerId (long) – ID do jogador
- balance (long) – balanço de pontos do jogador

- lastUpdateDate (date time) – data em que o balanço de pontos do jogador foi atualizado pela última vez (usando o fuso horário do servidor de aplicação)

Requisito Opcional 1

Seu gerente mencionou que seria ótimo se conseguíssemos gravar no banco de dados quando o balanço de pontos do jogador se tornou negativo.

Requisito Opcional 2

Adicione um serviço que também mostre um placar dos 100 melhores jogadores baseado no número de jogos realizados.

Requisito Opcional 3

O placar se tornou tão popular que o número de jogadores cresceu muito, tornando inviável que um único servidor dedicado consiga dar conta da carga. Como nós podemos escalar esse serviço horizontalmente? Descreva como atingir isso, não é necessário codificação.

Nota

Os jogadores são muito competitivos, e há vários jogadores ativos ao mesmo tempo, então o endpoint do placar de classificação será chamado várias vezes (milhares de requisições por minuto). Ainda não se sabe o quão valiosa essa funcionalidade será para o negócio da empresa, então inicialmente isso irá rodar como um piloto em um servidor dedicado que precisa lidar com toda a carga.

Boa sorte ;)