

API Rest

- API Rest
 - Retorno
 - Dados retornados
 - end-points
 - biblioteca
 - loja
 - comprar_jogo
 - ofertas
 - trocas
 - propostas
 - comprar_oferta
 - aceitar_troca
 - session
 - validar_jogo
 - jogo
 - user_info
 - avaliacoes
 - aumentar_saldo
 - historico
 - gerenciar_jogos

Retorno

A API sempre retorna uma tupla (`bool`, `None|str|{}|[]`), sendo o primeiro elemento usado para indicar se a execução foi bem sucedida, o segundo elemento ou retorna motivo do erro, caso algum erro tenha ocorrido, ou conforme especificado abaixo.

Dados retornados

- **BIBLIO_INFO**: {"id_chave": int, "avaliacao_usuario": float, "disponibilidade": str, ...JOGO};
- **JOGO**: {"id_jogo": int, "nome": str, "descricao": str, "avaliacao": float, "preco": float, "tags": list[str], "link_imagens": str, "link_trailer": str};
- **OFERTA**: {"id_oferta": int, "data": timestamp, "usuario_rank": float, "preco_oferta": float, ...JOGO};
- **TROCA**: {"id_troca": int, "data": timestamp, "usuario_rank": float, ...JOGO};
- **PROPOSTA**: {"id_proposta": int, ...JOGO};
- **JOGO_PENDENTE**: {"rank_dev": "float": int, ...JOGO};
- **AVALIACAO**: {"avaliacao_usuario": float, "resenha": str, ...JOGO};
- **HISTORICO**: {"id_transacao": int, "id_usuario": int, "id_usuario_participou": int, "id_chave": int, "valor": float, "tipo": char, "direcao": char, "data": timestamp};
- **USER_ROLE**: 'user'|'dev'|'admin';

end-points

biblioteca

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[BIBLIO_INFO, ...]</code>	Obtém uma lista com todos os jogos pertencem ao usuário	Deve estar logado, usa sessão para inferir id do usuário

loja

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[JOGO, ...]</code>	Obtém uma lista com todos os jogos que podem ser comprados diretamente da loja	-

comprar_jogo

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"id_jogo": int}</code>	<code>{}</code>	Compra jogado diretamente da loja	Deve estar logado, usa sessão para inferir id do usuário

ofertas

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[OFERTA, ...]</code>	Obtém uma lista com todos os jogos que foram colocados em oferta	-
POST	<code>{"id_chave": int, "preco": float}</code>	<code>{}</code>	Coloca um jogo da biblioteca para oferta com o preço <code>preco</code>	Deve estar logado, usa sessão para inferir id do usuário
DELETE	<code>{"id_oferta": int}</code>	<code>{}</code>	Remove um jogo ofertado	Deve estar logado, usa sessão para verificar se usuário é o criador da oferta

trocas

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[TROCA, ...]</code>	Obtém uma lista com todas as trocas disponíveis	-
POST	<code>{"id_chave": int}</code>	<code>{}</code>	Coloca um jogo para troca	Deve estar logado, usa sessão para inferir id do usuário
DELETE	<code>{"id_troca": int}</code>	<code>{}</code>	Remove a troca	Deve estar logado, usa sessão para verificar se usuário é o criador da troca

propostas

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{"id_troca": int}</code>	<code>[PROPOSTA, ...]</code>	Obtém uma lista com todas as propostas recebidas para determinada troca	Deve estar logado, usa sessão para verificar se usuário é o criador da troca
POST	<code>{"id_chave": int, "id_troca": int}</code>	<code>{}</code>	Coloca um jogo como contrapartida	Deve estar logado
DELETE	<code>{"id_proposta": int}</code>	<code>{}</code>	Remove a contrapartida	Deve estar logado, usa sessão para verificar se usuário é o criador da troca ou da proposta

comprar_oferta

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"id_oferta": int}</code>	<code>{}</code>	Compra jogo ofertado	Deve estar logado como usuário

aceitar_troca

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"id_troca": int, "id_proposta": int}</code>	<code>{}</code>	Aceita que uma troca seja feita com determinada contrapartida	Deve estar logado, usa sessão para verificar se usuário é o criador da troca

session

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"login": str, "password": str}</code>	<code>USER_ROLE</code>	Cria nova sessão	-
DELETE	<code>{}</code>	<code>{}</code>	Encerra sessão	-

validar_jogo

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[JOGO_PENDENTE, ...]</code>	Obtém lista de jogos pendentes	Deve estar logado como gerenciador

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"id_jogo": int}</code>	<code>{}</code>	Aceita jogo	Deve estar logado como gerenciador
DELETE	<code>{"id_jogo": int, "justificativa": str}</code>	<code>{}</code>	Recusa/cancela um jogo	Deve estar logado, se desenvolvedor justificativa pode estar vazia, se gerenciador não pode estar vazio

jogo

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{"id_jogo": int}</code>	JOGO	Obtém informações sobre o jogo	-
POST	JOGO*	<code>{}</code>	Cria jogo	Deve estar logado como desenvolvedor
DELETE	<code>{"id_jogo": int, "justificativa": str}</code>	<code>{}</code>	Remove/recusa um jogo	Deve estar logado, se desenvolvedor justificativa pode estar vazia, se gerenciador não pode estar vazio

* Não deve conter os campos `id_jogo` e `avaliacao`;

user_info

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>{"login": str, "saldo": float, "rank": float}</code> , caso usuário; <code>{"login": str, "lucro": float, "rank": float}</code> , caso desenvolvedor;	Retorna informação do usuário	Deve estar logado como usuário ou desenvolvedor

avaliacoes

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	[AVALIACAO, ...]	Recebe todas as avaliações feitas, caso usuário; Recebe todas as avaliações recebidas, caso desenvolvedor;	-

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{"id_jogo": int, "nota": float, "resenha": str}</code>	<code>{}</code>	Faz uma avaliação para um jogo	Deve estar logado como usuário

aumentar_saldo

Método	Argumentos	Resposta	Descrição	Restrição
POST	<code>{" acrescimo": float}</code>	<code>{}</code>	Aumenta saldo do usuário	Deve estar logado como usuário

historico

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[HISTORICO, ...]</code>	Caso usuário, retornas todas transações que fez parte; Caso desenvolvedor, retorna todos transações em que seu jogo foi comprado; Caso gerenciador, retorna todas transações concluídas no sistema;	Deve estar logado

gerenciar_jogos

Método	Argumentos	Resposta	Descrição	Restrição
GET	<code>{}</code>	<code>[JOGO, ...]</code>	Retorna todos os jogo do desenvolvedor que estão atualmente na loja.	Deve estar logado como desenvolvedor