

Projektname: The Wallstreet Wizzard – Demo Brocker

Anhang GraphQL Documentation

Verfasser*innen:

Marko Robert Szönyi, Arvid Harang, Lennart Pfeiler

Studiengang Wirtschaftsinformatik

Bearbeitungszeitraum: September 2024 – Oktober 2024

Matrikelnummern:

2045018

3959770

6667860

Kurs: WI23A

E-Mails:

mar.szoenyi.23@lehre.mosbach.dhbw.de,

arv.harang.23@lehre.mosbach.dhbw.de,

len.pfeiler.23@lehre.mosbach.dhbw.de

Studiengangsleiter: Prof. Dr. Christoph Sturm

GraphQL Documentation

Name	Field	Type	Description
Query	getUserProfile	(email: ID!, token: String!)	Ruft das Benutzerprofil ab
Query	getStock	(email: ID!, token: String!, symbol: String!)	Ruft Informationen zu einer bestimmten Aktie ab
Query	getAllPortfolioStocks	(email: ID!, token: String!, sortBy: String)	Ruft alle Aktien im Portfolio des Benutzers ab
Query	getPortfolioStock	(email: ID!, token: String!, symbol: String)	Ruft eine bestimmte Aktie im Portfolio ab
Query	getAllTransactions	(email: String!, token: String!, sortBy: String!)	Ruft alle Transaktionen des Benutzers ab
Query	getUserPortfolio	(email: ID!, token: String!)	Ruft das Portfolio des Benutzers ab
Mutation	createUser	(firstname: String!, lastname: String!, email: ID!, password: String!, budget: Float!)	Erstellt einen neuen Benutzer
Mutation	editProfile	(currentToken: String!, currentMail: String!, firstname: String, lastname: String, email: String, password: String, budget: Float)	Bearbeitet das Benutzerprofil
Mutation	resetProfile	(SessionEmail: ID!, token: String!)	Setzt das Profil zurück
Mutation	deleteProfile	(SessionEmail: ID!, token: String!)	Löscht das Benutzerprofil
Mutation	createStock	(email: ID!, token: String!, symbol: String!, stockPrice: Float!, name: String!)	Erstellt eine neue Aktie in der Datenbank
Mutation	editCurrentValue	(token: String!, email: String!, symbol: String!, newValue: Float!)	Ändert den aktuellen Wert einer Aktie
Mutation	addPortfolioStockOrder	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email:	Fügt eine neue Transaktion im Portfolio hinzu

		String!, symbol: String, token: String!)	
Mutation	createBuyOrder	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email: String!, symbol: String, token: String!)	Erstellt eine Kauforder für eine noch nicht vorhandene Aktie
Mutation	increasePortfolioStockOrder	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email: String!, symbol: String, token: String!)	Erhöht die Position einer bestehenden Aktie im Portfolio
Mutation	decreasePortfolioStockOrder	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email: String!, symbol: String, token: String!)	Verkauf einer Aktie um eine bestimmte Position
Mutation	deletePortfolioStockOrder	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email: String!, symbol: String, token: String!)	Löscht eine bestimmte Transaktion
Mutation	handleAlexaRequest	(alexarO: AlexaROInput!)	Handhabt eine Anfrage von Alexa, basierend auf den Eingaben
Input	AlexaROInput	response: ResponseInput, session: SessionROInput	Strukturierte Eingabe für Alexa-Anfragen.
Input	ResponseInput	outputSpeech: OutputSpeechInput, shouldEndSession: Boolean	Eingabe zur Definition der Antwortstruktur für Alexa
Input	OutputSpeechInput	(type: String, text: String)	Strukturierte Eingabe für die Alexa-Sprachausgabe

Input	SessionROInput	attributes: [SessionAttributeInput]	Sessioninformationen für Alexa-Anfragen
Input	SessionAttributeInput	key: String, value: String	Sessionattribute für Alexa-Anfragen
Output	User	firstname: String, lastname: String, email: ID!, password: String, budget: Float	Benutzerinformationen
Output	Portfolio	portfolioID: ID!, value: Float, startValue: Float, email: ID!	Details zu einem Benutzerportfolio
Output	Transaction	transactionID: ID!, transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float, totalPrice: Float, email: ID!, symbol: String, leftInPortfolio: Float	Transaktionsdetails
Output	PortfolioStock	portfolioID: ID!, symbol: String, stockAmount: Float, boughtValue: Float, currentValue: Float	Details zu einer Portfolio-Aktie
Output	Stock	symbol: String, stockPrice: Float, name: String, sortBy: String	Aktieninformationen
Output	StringAnswer	answer: String	Eine einfache Antwort als Zeichenkette
Output	AlexaRO	response: Response, session: SessionRO	Strukturierte Antwort für Alexa-Anfragen
Output	EditRequest	(currentToken: String!, currentMail: String!)	Anfrage zur Bearbeitung des aktuellen Profils
Output	Session	(token: String, SessionEmail: ID!)	Repräsentiert eine Benutzersession
Output	EditCurrentValue	(token: String!, email: String!, symbol: String!, newValue: Float!)	Anfrage zur Aktualisierung eines Wertes
Output	StringAnswer	(answer: String)	Antworttyp mit einer Zeichenkette
Output	TransactionContent	(transactionType: Int, stockAmount: Float, date: String, pricePerStock: Float,	Repräsentiert den Inhalt einer Transaktion

		totalPrice: Float, email: ID!, symbol: String)	
Output	OutputSpeech	(type: String, text: String)	Repräsentiert die Ausgabesprache für Alexa
Output	SessionRO	(attributes: [SessionAttribute])	Repräsentiert die Sessionattribute
Output	SessionAttribute	(key: String, value: String)	Repräsentiert ein Attribut in einer Session
Output	IntentRO	(name: String, slots: [Slot])	Repräsentiert eine Absicht in einer Alexa-Anfrage
Output	Slot	(name: String, value: String)	Repräsentiert einen Slot in einer Alexa-Anfrage
Output	Response	(outputSpeech: OutputSpeech, shouldEndSession: Boolean)	Repräsentiert die Antwort von Alexa
Union	UserResult	User, StringAnswer	Kombination aus Benutzerinformationen und String-Antwort
Union	StockResult	Stock, StringAnswer	Kombination aus Aktieninformationen und String-Antwort