

Margin Service API

The Margin Service API is designed to give all market participants direct access to CME Clearing initial margin engines for portfolios. It allows users to pass through portfolios of all CME Cleared products for initial margin and incremental margin calculation and provides features such as Interest Rate Portfolio Margin savings and margin estimation.

Use this search bar to search topics within the Margin Service API.

This information is intended to provide a complete overview of the Margin Service API for development purposes. To discuss certification testing and the licensing process, please see [Onboarding and Verification](#).

[Contact us](#) for any onboarding or production support.

This page describes:

- [Features](#)
- [Margin Service API Versions and Latest Schema](#)
- [Technology](#)
 - [REST](#)
 - [Asynchronous vs. Synchronous Requests](#)
 - [Contact](#)

Related Content

- [Margin Service API - Analytics](#)
- [Margin Service API - Authentication and Entitlement](#)
- [Margin Service API - Developer Guide](#)
- [Margin Service API - Onboarding and Verification](#)
- [Margin Service API - Requests](#)
- [Margin Service API - REST](#)
- [Margin Service API - Types](#)
- [Margin Service API - Use Cases](#)
- [Risk API](#)

Features

The Margin Service API supports the following features:

- Portfolio maintenance
- Margin calculation
- Optimize calculation
- Incremental margin
- Analytics (portfolio margining)
- Margin estimation and full revaluation for OTC Interest rate swaps
- [Margin Service API ALL-IN-ONE call](#)
- [What-If Analysis](#)
- Futures Compression (requires separate license)

The Margin Service has 6 main endpoints: Analytics, Margin, Optimize, Portfolio, Transactions, and All-in-One. Each endpoint supports the standard GET, POST, PUT, and DELETE HTTP functions.

Margin Service API Versions and Latest Schema

Margin Service API supports concurrent versions. An XML schema is provided for validation of XML requests and responses supported by the API.



For best results, use **right click + Save As** to download each file.

Version /Schema	Download Link	Additional API Features	NR Release Date	Production Release Date
1.0	Download 1.0	Original Version	Original	Original
1.1	Download 1.1	All-In-One Call	Nov 17, 2013	Nov 17, 2013
1.2	Download 1.2	What-If Analysis	Dec 4, 2013	Jan 20, 2014
1.3	Download 1.3	Swaptions-specific reporting	Mar 2, 2016	Apr 11, 2016
1.4	Download 1.4	Real-time Margin over API Margin Matrix over API	July 6, 2016	July 18, 2016
1.5	Download 1.5	Futures Compression (no margin functionality)	August 14, 2018	August 24, 2018
1.6	Download 1.6	Optimizer over API	November 27, 2019	December 14, 2019

1.12*	Download 1.12	SPAN 2 Features (quick documentation links below): <ul style="list-style-type: none"> • Transaction - Add (see Futures and Options CSV and JSON examples) • Margin - Calculate (see "Request - SPAN 2 Margin") • Margin - Get (see "Response for Futures and Options - when riskFramework = NEXT") • ALL-IN-ONE Call (see "JSON All In One Request" example) 	June 15, 2021	June 28, 2021
-------	-------------------------------	--	---------------	---------------

*Latest Version

Minor (.X) API versions are backward compatible. We ask that all users update to the latest version of the Margin API schema at their earliest convenience.

For all POST/PUT calls, the core schema version is auto-detected based on the xmlns specified in the root element of the request; the response will be returned in the same schema version.

For all GET/DELETE calls, the API will return the latest schema unless the version is specified as a request parameter in the URL (e.g. /MarginServiceApi/1.1/portfolios/\${marginPortfolio}).

Best Practice is to specify the version on every request: (e.g. /MarginServiceApi/1.1/portfolios/\${marginPortfolio})

The New Endpoints for the Schema 1.3 are:

Transactions

/MarginServiceApi/1.3/transactions

Margin

/MarginServiceApi/1.3/margins

Poll Margin

/MarginServiceApi/1.3/margins/{post margin response id}

The New Endpoints for the Schema 1.4 are:

Margin Matrix:

/MarginServiceApi/1.4/analytics/margin-matrix/us-swaptions-buy

/MarginServiceApi/1.4/analytics/margin-matrix/us-swaptions-sell

/MarginServiceApi/1.4/analytics/margin-matrix/us-swaptions-straddle

/MarginServiceApi/1.4/analytics/margin-matrix/us-vanilla

/MarginServiceApi/1.4/analytics/margin-matrix/us-ois

Real-time Margin:

/MarginServiceApi/ 1.4 /Analytics/RealTimeMargin

The New Endpoints for the Schema 1.6 are:

Post Optimize

/MarginServiceApi/1.6/optimize

Get Optimize

/MarginServiceApi/1.6/optimize/{post optimize response id}



Notes:

The default if no schema is provided is always 1.2

Schema 1.0, 1.1, 1.2 are supported

Schema 1.3 is required for Swaptions

Schema 1.4 is required for Real-time Margin and Margin Matrix over API

Schema 1.5 is required for Futures Compression

Schema 1.6 is required for Optimizer over API

Schema 1.12 is required for SPAN 2 margin

Best practice is to call the API using latest which returns the latest schema.

Example:

`/MarginServiceApi/latest/margins/id`

Technology

REST

The Margin Service API is built upon [Representation State Transfer \(REST\)](#) web service technology. In this methodology HTTP URLs and methods (such as `GET`, `POST`, etc.), parameters, and statuses are used to model the remote procedure call behavior.

The API utilizes a custom XML language for requests and responses, which can contain contents in [various formats](#), including industry standard XML formats such as FpML and FIXML. Schemas have been provided for validating the various [request and response](#) XMLs which are possible in the API.

- View [sample messages and error messages](#)
- See [Wikipedia](#) for a full discussion of REST technology

Asynchronous vs. Synchronous Requests

For requests which are expected to be executed quickly, the API will return the results of the request in the action (synchronously). For some requests (such as margin calculation) which may take longer to process, an asynchronous model is provided. After submitting the request, the service will return an identifier which can be used to query for the results of the operation with subsequent requests. It is then up to the client to poll the server for completion of the request

Contact

For more information and questions please contact the [support team](#).