NMFTA Rate Quote API Product Requirements

Overview

Rate Quote is an API capability that delivers spot freight rates (dynamic market prices) to shippers in real-time, enabling automated rate shopping and booking for freight shipments. This is part of the NMFTA's Digital Standards initiative, particularly under the Full Truckload (FTL) Digital Council, aiming to standardize how pricing information is exchanged via open APIs. By providing instant quotes for various shipment types (Full Truckload, etc.), the Rate Quote API eliminates the delays of manual rate inquiries and allows shippers to quickly secure competitive prices for their loads. The rates are dynamic, reflecting current market conditions such as capacity and demand, and they cover different equipment types (e.g., dry van, flatbed, refrigerated). This means a shipper can just as easily get a spot quote for a full 53' truckload, through a unified communication protocol.

By implementing this API, all parties – shippers, carriers, and intermediaries – share a common method for accessing spot rates. Rate quotes, when combined with other NMFTA standard APIs (e.g. electronic tendering and eBOL), form a continuous digital thread from load creation to completion, greatly enhancing transparency and efficiency in freight movements.

Value Proposition

- **Shipper**: The Rate Quote API allows shippers to request spot rates from multiple carriers within their network or even outside of it, ensuring they obtain the most competitive rate for each load. Instead of phone calls or emails to brokers and carriers, a shipper can instantly compare pricing, transit times and service offerings. Moreover, by automating the quote retrieval, shippers can integrate this into their transportation management systems (TMS) or even e-commerce platforms to provide immediate shipping cost estimates to customers. The time savings and improved visibility to current market prices empower shippers to respond faster to capacity crunches and budgetary constraints. Additionally, receiving the quote can reduce invoice disputes that take time to resolve after the shipment has occurred.

- **Carrier**: Carriers benefit by digitally offering real-time spot rates based on their available capacity, increasing their chances to win freight in the spot market. The API standardizes how carriers receive quote requests and respond, reducing the overhead of negotiating each load. Carriers can adjust pricing algorithmically using their asset availability and desired lane yields, and respond via the API within seconds. This opens opportunities to fill backhauls or last-minute capacity and can lead to higher asset

utilization. Each quote can be time-bound – for example, a carrier might specify that a rate is valid for 24 hours or until a certain date – after which it can expire if not accepted. (The NMFTA standard envisions that quote validity is configurable; a typical default is that rates remain valid up until the scheduled pickup date unless conditions change.) By simplifying the spot quote process, carriers can handle more quote volume and potentially secure more business.

- **3rd Party:** The Rate Quote API will allow third parties of all kinds to receive and send necessary shipment information to ensure a seamless flow through the life cycle of a shipment. Due to the nature of 3rd parties business and having to interact with multiple parties in the transportation industry, 3rd parties stand much to gain by simplification of their business processes, which the Rate Quote specification will enable. Overall, the Rate Quote standard helps 3rd parties coordinate more effectively and provide better service with less overhead.

Overall, the Rate Quote API accelerates the freight booking cycle: a shipper with a new load can retrieve a price and, if acceptable, immediately tender the load at that price, all electronically. It reduces uncertainty by providing transparency into current market rates and ensures that all parties (shipper, carrier, broker) are working off the same timely pricing information. This API complements other standard APIs (for tendering, tracking, billing, etc.) to move the industry toward a fully digital quote-to-cash process – for example, a shipper might use the Rate Quote API to get a price, use a standard tender API to book the shipment at that price, then later use tracking APIs to monitor it, and so on, with each step integrated and efficient.

Definitions

The following terms are used in the Rate/Quote specification. Definitions align with the NMFTA Digital Standards Development Council (DSDC) Glossary of Terms:

Application Programming Interface (API): An API is a set of rules using HTTP that enables software applications to communicate with each other, exchanging data, features, and functionality. In transportation, APIs can be used to integrate systems for tracking shipments, managing logistics, and automating processes.

Spot Rate / Spot Quote: A one-time freight rate for a specific shipment, offered based on current market conditions. Spot rates are typically used for non-contracted (ad hoc) loads. They can fluctuate day-to-day or even hour-to-hour with the market. In this context, a “Rate Quote” usually refers to a spot rate retrieved via the API for a given origin, destination, and time frame. The quote is time-sensitive – it may only be valid for a certain period or until the shipment’s pickup date – and if not accepted in time, it may expire or need to be refreshed at a new rate.

Contract Rate: (Mentioned for context) A freight rate that is part of a long-term agreement between a shipper and a carrier (e.g., annual contract). Contract rates are typically static or adjusted periodically and are not the primary focus of this API. However, some implementations might allow retrieval of contract rates through the same interface if authenticated as a particular customer.

Full Truckload (FTL): Full Truckload refers to a transportation mode where a truck carries a Load for a single customer, meaning the journey is reserved for that customer only. This is typically used for large Shipments that can fill an entire truck.

Less-Than-Truckload (LTL): Less Than Truckload is a shipping service for smaller freight loads that combines shipments from multiple customers to optimize costs and efficiency. It is used when shipments do not require the full capacity of a truck.

Volume LTL: A shipment that is technically LTL (i.e., not an entire truckload) but large enough to qualify for special LTL pricing, often somewhere between 6 and 12 pallets or over ~5,000 lbs. Many LTL carriers offer “volume” or “spot quote” rates for these shipments because they don’t fit neatly into the standard class tariff. The Rate Quote API specifically mentions support for Volume LTL options. This means if a shipment’s size is beyond typical LTL, the request can indicate volume LTL and carriers may respond with a discounted rate that is lower than the sum of normal LTL pricing but higher than a full truckload. In practice, the user might set a flag or simply input a large size, and the system will return a volume LTL quote.

Accessorial: Accessorial services in freight are additional services beyond standard dock-to-dock transport – e.g., liftgate delivery, residential pickup, limited access location, inside delivery, appointments, etc. These often incur extra fees. In the context of Rate Quote API, accessorials can usually be specified in the request (for example, indicating “LIMITED_ACCESS” for pickup or “LIFTGATE” for delivery). Carriers that receive these indicators will include the cost of those services in their quoted rate. Many accessorials are especially relevant for LTL. The API standard includes fields for location-level requirements (like whether a stop requires a liftgate or has an appointment) so that the returned quote accounts for those needs. It’s worth noting that some special accessorials may not be fully supported in automated pricing – for instance, very unusual requests might return a note like “call for quote” or simply not be included in the automated rate. (One source noted that the transactional pricing engine focuses on linehaul and fuel, and might not include every accessorial charge in the upfront quote, especially for truckload.)

Refrigerated (Reefer): A container designed and equipped for the transportation of goods that need to be temperature controlled during shipping. In quoting, a reefer is considered a distinct equipment type from a dry van. Reefer shipments also have parameters like required temperature range.

Equipment Type: Physical assets used for storing and transporting freight. Common types include Dry Van (default for truck), Refrigerated Van (Reefer), Flatbed, as well as specialized equipment like step-deck, doubles, etc. Equipment characteristics (such as height, length) might also be specified; e.g., some systems allow indicating if a 53' or 48' trailer is needed, or if tarps are required on a flatbed. These factors can affect the rate or the eligibility of carriers to respond.

Quote ID: A unique identifier assigned to a specific rate quote returned by the API. This ID allows the quote to be referenced in subsequent operations. For example, if the shipper decides to book the shipment, they might pass the Quote ID along with the booking request so the carrier can tie the booking to the quote and honor that rate. The Quote ID is typically a GUID or string that is globally unique. It may encode some information internally (like carrier or timestamp) but is opaque to the API user. In practice, Quote IDs are valid only for the lifespan of the quote (until expiration) and become irrelevant after the load is booked or the quote expires.

Platform (in Quote Context): In some systems, "platform" refers to the channel or environment from which the quote request originates. For example, a large 3PL or a managed transportation platform might have an assigned platform ID when interfacing with a carrier's API. The platform identifier can route the quote request to specific business logic. This is more of an implementation detail: the NMFTA standard provides a field for such purposes so that if, say, a 3PL has a special arrangement or logic on the carrier side, the platform ID tells the carrier's system to apply that. It's not something the average shipper uses, except those who have been given one by the carrier (e.g., some carriers require TMS vendors to send a platform code for analytics or differentiated pricing).

Authentication: (Not a specific term in the documents, but relevant concept) An API key or token is used to identify the calling party. For instance, if accessed as a guest (no authentication), the quote might be returned using a generic account with limited privileges. If authenticated as a specific shipper, the API might return contract rates or apply the shipper's discounts if applicable. In all cases, to actually book a quote, authentication is needed. For the scope of the product requirements, it's assumed that secure access will be in place (usually via API keys and/or OAuth tokens).

These definitions ensure a common understanding of key terms used in the Rate Quote API specification and align with industry-standard terminology.

Best Practices

When implementing or using the Rate Quote API, it's important to understand which data fields are required vs. optional and how to maximize the value of the shared information. The following best practices should be observed for an effective implementation:

- Minimum Shipment Details: To get the most accurate rate, provide as much information as possible about the shipment beyond the minimum. At a minimum, mode of transport (LTL vs FTL), origin and destination (at least ZIP/postal codes), and pickup date are required for any quote.

- Possible Optional Shipment Details: However, adding details like number of stops, the full street address (which can affect things like terminal routing or remote area surcharges), the delivery date or required transit time (if you have a deadline), hazmat in or descriptive information about the freight will refine the quote. For example, including the commodity weight and dimensions for an LTL shipment can allow the API to compute if the shipment is density-based class and potentially qualify for a volume LTL rate. If you only provide weight and omit pallet count or class for LTL, the system might assume a worst-case class, resulting in an inflated quote or an error. Rule of thumb: always include origin, destination, pickup date, and all commodity details (weight, pieces, class or density) in the request for LTL; for FTL, include any special requirements (like "needs a refrigerated trailer at 34°F" or "flatbed with tarps") so that "the returned quote is applicable.

- Consider Modes: The API is designed to support both Full Truckload (FTL) and Less-than-Truckload (LTL) quoting. Requesters may explicitly indicate the desired mode (e.g., FTL or LTL), or omit it to receive quotes for all applicable modes based on the shipment details provided. Implementation approaches vary: some platforms use a simple mode flag or separate endpoints (e.g., /quote/ltl vs. /quote/tl), while others infer the mode from the request data—such as identifying LTL from the presence of pallet dimensions, weight, and NMFC class. Developers should structure their implementations to accommodate and parse multiple quote options, ensuring flexibility and adaptability to platform-specific logic.

- Use Correct Units and Codes: Make sure to format data according to the API specification. Use proper country and postal codes for locations (e.g., "USA" for United States in ISO3 format, and 5-digit ZIP for US zips).
  The scope will be limited to United States, Canada and Mexico (where it aligns with US and Canada Standards). List weights in the required unit (usually pounds for US-based APIs unless specified otherwise) and dimensions if needed (often inches or feet). If

specifying freight class, use the numeric class or NMFC code as expected. For example, class “125” should be numeric rather than text. If the API allows a platform or customer code, ensure you include the one provided to you; this could route your request through a custom pricing logic meant for your account. Also, follow the date/time format exactly (often ISO 8601). Minor format issues (like a wrong timezone format or missing country code) can lead to a rejected request.

- Indicate Accessorial and Special Requirements: Optional fields exist for a reason – using them can save headache later. If your pickup or delivery has any special requirements (limited access, need a liftgate, appointment required, residential address, etc.), indicate these in the quote request. Why is this important? Because the quote returned should include any extra fees for those services. If you fail to mention, say, that the delivery is residential, the quote might come back $300, but later when the carrier finds out it’s residential, they’ll add a $100 surcharge, causing discrepancy. By including it up front, you might instead get a quote of $400 that already accounts for residential service, giving you transparency. Similarly, for full truckload, if extra services like team drivers (for expedited transit) or tarping on a flatbed are needed, indicate them (many APIs have a field like specialServices where you can list “teamDriver” or “tarping”). This ensures the carriers that can meet those needs respond, and they factor any cost premiums into the rate.

- Understand Pricing Factors: The simplest inputs for a quote are origin, destination, and weight (for LTL) or mileage (for TL). However, real-time pricing engines often consider a limited set of factors. For example, one digital pricing engine notes that for truckload spot quotes, it really only considers origin/destination (postal codes), shipment date, and total weight (plus basic mode/equipment) – other details like pallet count, exact commodity type, etc., typically do not affect a pure linehaul spot quote. What this means: don’t overly complicate a request with extraneous data expecting the price to change. If you add five different reference numbers or a long description of goods, it won’t change the rate; it’s only for your or the carrier’s reference. On the other hand, omitting key data that does impact price is dangerous. For instance, not mentioning that a load is hazmat when it is (many carriers exclude hazmat from dynamic quotes). In fact, if you need a hazmat shipment quote, you might have to accept that the API could return an error or no rate, and you’ll need to contact the carrier’s rep. Similarly, extremely high-value loads or those requiring high liability coverage might not get an auto-quote (or will exclude insurance in the quote). In summary, provide the key inputs that affect price (weight, class, distance, mode, special services) and know that other inputs are either for information or not processed by the automated engine.

- Multi-Stop and Complex Moves: If your shipment has multiple pickups or deliveries (multi-stop truckload), use the provided structure to list intermediate stops in order. The Rate Quote API can handle multi-stop quotes, but you must include each stop’s location and indicate whether it’s an extra pickup or delivery. Each extra stop usually adds a cost

(stop-off charge) and additional mileage. Some carriers may not provide an automated quote for multi-stop TL if it's very complex; others will. Make sure to also include any stop-level requirements (e.g., stop 2 needs an appointment). The quote response should then reflect the entire route's cost. If the API cannot handle multi-stop (some simpler ones only do point A to point B), you might have to obtain separate quotes or contact for a manual quote. The NMFTA standard supports multi-stop in the schema, so ideally systems implement it. For LTL, multi-stop doesn't apply (each LTL shipment is one origin, one destination typically).

- Leverage "Flexible Dates" Option If Available:
  Quoting systems allow a shipper to say, "I'm flexible on pickup (or delivery) by a certain number of days – give me the cheapest rate in that window." If the Rate Quote API implementation you're using has a flexibleDates or similar parameter, consider using it if your pickup date is not fixed. For example, setting flexibleDays: true might instruct the engine to look at the 3-day window around your requested pickup and return the lowest-cost option (maybe picking a day where a backhaul is available). This can result in significant savings if your schedule permits. The response would likely indicate the date of the cheapest option. If you have a hard date, keep this false. By default, assume quotes are for the exact date given, unless you explicitly ask for flexibility.

- Review Quote Validity: When processing a quote response, review the terms that indicate how and when the quote applies. In many cases, a quote is considered valid through the pickup date provided in the request—especially for future-dated shipments. Some systems may explicitly state the applicable conditions or timeframes within the response. It's important to understand how the quoting system defines validity so downstream processes (like booking) align correctly with the quote's scope and assumptions.

- Quote Expiration: If the quote response includes a field such as validIfTenderedBefore or another expiration indicator, systems should be designed to capture and evaluate this information programmatically. Rather than relying on manual processes or human intervention, platforms should surface expiration details to users and enforce logic that prevents booking against expired quotes. Automating the handling of expiration windows ensures consistency, reduces errors, and supports a more reliable end-to-end digital workflow.

- Requote: In cases where shipment details change or too much time has passed since the original quote was received, it's best practice to request a new quote via the API rather than relying on the previous quoteId. The API is built for fast, repeatable calls, allowing users to retrieve updated rates quickly—even minutes or hours after the initial request. This flexibility is especially important in dynamic or volatile pricing environments, where rates can fluctuate frequently.

- Error Handling and Fallback: Be aware of what to do if the Rate Quote API returns no rate or an error for a request. A “rate not found” scenario could happen for several reasons: no carrier in network can service that lane, the shipment violates some constraint (too heavy, too large, hazardous, etc.), or simply a technical issue. Best practice is to build logic that if no quote is returned, you either try adjusting the request (e.g., ensure all required fields are there, or remove an unusual accessorial to see if that was blocking it) or have a secondary process (like sending an email to carriers or triggering a spot bid request manually). Typically, the API will return a specific error message if something is wrong with the request (e.g., missing postal code = 400 Bad Request with detail). Those should be fixed in code. If the request is fine but no carrier offers a rate, you might get a response that simply says rateFound: false or an empty list of options. Handle this gracefully – e.g., notify the user “No automated rate available, please contact support for a manual quote.” Also, log or surface any warnings or remarks the API gave. For instance, some responses include a “remarks” field with reasons like “No capacity – try again later” or “Invalid lane (service not available)”. Use those clues to decide next steps.

- Security and Throttling: On the technical side, ensure protection of API credentials and use them as intended (usually in an HTTP header). Follow the carrier or provider’s guidelines on how often you can call the Rate Quote API. While you might be tempted to poll rates continuously to sense market trends, providers may have usage limits. As a courtesy and best practice, only request quotes when you need them (e.g., when you have a real shipment to price). If you want to do extensive rate discovery (say hundreds of lanes for budgeting), coordinate with the provider – they might have a bulk quoting endpoint or offline tool. Overuse of the live API could result in throttling (HTTP 429 Too Many Requests). The NMFTA standard will ensure compatibility across platforms, but individual carriers might extend fields (like adding a new equipment type). Staying up-to-date will let you take advantage of new features.

- Quote-to-Order Integration: Design the process such that once you get a quote and decide to book it, seamlessly transitioning to tendering the load. Ideally, the system would take the Quote ID and chosen option and pass it to the booking (tender) API, thus avoiding re-keying and ensuring the carrier knows which quote you’re accepting. Many carriers have an operation like “book quote” or they simply expect the Quote ID in the tender request. Follow those practices to lock in the rate. If your system cannot do this automatically, at least present the necessary info to the user making the booking so they can reference it. Also, store the quote details (rate, carrier, etc.) as a record – this helps later to audit freight invoices. Carriers have indicated that if a load is tendered referencing a quote, they will honor that price barring extraordinary differences in what’s moved. So linking the two is critical to avoid billing discrepancies.

By adhering to these best practices, you can maximize the benefits of the Rate Quote API: you'll get accurate pricing with no surprises, you'll reduce the need for follow-up communications, and you'll integrate the quoting step smoothly into your freight execution workflow. In short, be thorough and precise in your quote requests, and the automated system will return the best it has to offer.

Requirements

Request Quote: Request

As a Shipper or 3PL, I want to obtain a real-time rate quote for a planned shipment by providing key shipment details to the API.

- In a Rate Quote request, the client (shipper, 3PL, etc.) MUST provide basic information about the shipment:
    - Origin location Zipcode
    - Destination location Zipcode
    - Requested pickup date range
    - Equipment Type
    - Mode(FTL)
  - These are non-negotiable fields; without them, an accurate rate cannot be determined. Origin and destination are typically specified by postal code (and country code if international) at minimum, but providing full city/state or street address details is supported and recommended. The pickup date can be a specific date or date-time; it should be realistic (e.g., not in the past, and usually not more than 45 days in the future as many systems limit how far out you can quote). If a pickup time window or an appointment time is relevant, the request can include that – otherwise, just a date is sufficient (it's assumed all day availability unless stated).

- The client MUST identify the mode they are interested in. This can be implicit or explicit.
    - TL
    - LTL.
  - By default, one should assume a single mode per request.
  - If both LTL and TL options are desired for comparison, the client might need to call the API twice (once in each mode) – or call a combined endpoint that returns both.

- The client MUST provide shipment characteristics relevant to rating.
    - Total Weight
    - Trailer Type
  - A value of 0 or none will likely result in a bad request or a very unreliable quote. If the shipment is an unusual shape or size (e.g., over-dimensional for a truckload,

or requires a flatbed), those details must be included via the equipment type or special instructions.
    - For hazardous materials, the client MUST indicate if the freight is hazardous (hazmatFlag = true and provide relevant hazmat details), and be prepared that some carriers won't quote it automatically.
    - (Context only) For LTL, it includes weight, dimensions or freight class, number of handling units (e.g., pallets), and commodity description.

- The client MAY provide additional optional details to improve accuracy:
    - Declared cargo value (especially if wanting insurance coverage info or if high value might restrict carrier selection),
    - Reference numbers (like a quote request ID on the shipper's side for tracking)
    - Desired equipment type (if requesting multiple quotes).
    - Weight
    - Volume
  - For example, if a refrigerated truck or container is needed, the client MUST set an equipment type field to "Reefer" (refrigerated) in the request; similarly, for a flatbed request, set it to "Flatbed."
  - Including an email or user ID in the request header is often required by carriers for logging purposes – e.g., the userEmail header must be set so the carrier knows who asked for the quote (some systems optionally use it to send quote confirmations or just for audit).

- If the API client has a specific billing account or customer code with the carrier, it MUST be provided in the request (often in an "options" object). This ensures that contract rates or specific pricing agreements are considered. If none is provided, the quote is given at public rates.

- Example (FTL Request): A shipper wants a dry van truckload quote from Chicago, IL 60601 to Dallas, TX 75201 for a pickup on 2025-08-01. They would send a request with origin postal code 60601 (USA), destination 75201 (USA), pickupDate of "2025-08-01", equipmentType "DRYVAN" (dry van is default), and maybe weight e.g. 20,000 lbs.

- Upon sending the request, the client will typically get an immediate acknowledgement (HTTP 200 OK with quote data, or an error if something is wrong). The design is synchronous – you send the request and get the quote(s) in the response.

In summary, the Create Quote request is analogous to filling out an online form for a freight quote – origin, destination, when, what is being shipped, and any special needs. The more correct and comprehensive the information provided, the more precise the returned quote will be, and the fewer adjustments will be needed later.

Request Quote: Response

As the Rate Quote system (carrier or platform), I want to return one or more rate quotes for the requested shipment, including all details necessary for the shipper to make a booking decision.

- In the response to a quote request, the system MUST indicate if a rate was found.
    - RateFound: True
    - RateFound: False
    - This is often a boolean field like rateFound or simply the presence of rate entries.
    - If no quote is available (no carriers could provide a rate), the system might return rateFound: false along with perhaps a message explaining why (e.g., "No service available for specified lane"). If the request was for TL and you're querying a broker network, you might also get multiple options (different carriers or modes).

- For each quote option, the response MUST provide core details, at minimum:
    - Price (cost)
    - Transit time or expected delivery
    - The price can be all-in, including fuel surcharge or any other assossorials, in the currency of the billing account (USD for domestic US typically).
    - It may be broken into components; e.g., lineHaulCharge and fuelSurcharge and then totalCharge.
    - The transit time might be given as an estimated number of days or a delivery date. For instance, the quote might say "Estimated Transit Days: 4" or provide a deliveryDate of Aug 10 for a pickup Aug 5.

- The response MUST include a Quote Identifier for each option.
    - QuoteId
    - If the shipper later wants to book that quote, they will refer to this ID.
    - It's typically a GUID or alphanumeric string unique across the system.
    - Even if only one quote is returned, an ID will be given. The system uses this to recall what rate was offered and ensure consistency when booking.
    - For systems that allow retrieving a quote later by ID, this is the key you'd use.
    - The requirement is that each quote option in the array has a distinct identifier and can be individually referenced.

- Optional Services: The system MUST account for optional services requested in the pricing. The requirement is that the totalCharge given truly reflects everything that was asked for, with no hidden add-ons later provided the information was accurate. The quoting system should clarify if the quote is firm or if any part of it is estimated/not included.

- Single-Load Assumption: The standard assumes that each quote request represents a single shipment or load. All responses should be treated as valid for one truck, one movement—not for multiple identical moves or repeated use. This ensures that carriers and 3PLs do not misinterpret a quote as permission to dispatch multiple trucks under the same rate or conditions. If additional loads are needed, separate quote requests must be submitted. Enforcing this single-load assumption helps maintain clarity, avoids overcommitment, and supports clean, auditable transactions across systems.

- Error Responses: If the quote request was malformed, the response will not be a quote but an error. For instance, if required fields were missing or had invalid values, the API would return a 400 Bad Request with a description of the error. This is not a quote per se, but it's part of the "response" possibilities. The system must handle these gracefully. From a requirements standpoint, any missing data error must clearly tell the user what was wrong (e.g., "Missing origin postal code" or "Invalid date format for pickupDate"). These error messages follow a problem details format often, enumerating each issue. That allows the calling application to fix the input and try again. Authentication errors (401) or service downtime (503) are also possible; those are system-level and should also be clearly indicated with standard codes.

- Monetary Units and Precision: All monetary values in the quote must be clear about currency. The API provides a currency code like "USD" with each monetary field. The values are often to two decimal places. The system should ensure no rounding issues – i.e., the total equals the sum of components exactly to the cent to avoid penny differences.

- Additional Metadata: The response could include a lot of metadata: for example, a quoteDateTime (timestamp when it was generated), a list of assumptions, or a reference to a tariff or quote record in the carrier's system. The product requirement is to be transparent: any factor that might be relevant to the shipper should be exposed. If the quote is subject to some minimum charge or capacity token, it should be mentioned.

- To illustrate, consider the earlier examples: the Chicago-Dallas FTL request might get a response like: Quote ID 12345, Carrier "ABC Trucking", mode "Truckload", equipment "53' Dry Van", transit 2 days, price $1,150.00 total (with fuel).

- And another option: Quote ID 67890, Carrier "XYZ FLATBED", mode "Flatbed", equipment "53' Container", transit 4 days, price $900.00 total.

The requirement is that the Rate Quote response provides all the necessary information to compare options and to select one for booking, without needing additional clarification. It should be as if a freight broker just emailed you a quote: you want to know price, who's moving it, how long it will take, and any special conditions. The API response must cover all that in structured form.

Retrieve Quote: Request (Optional)

As a Shipper or 3PL, I want to retrieve a previously generated quote by its ID in case I need to review it or use it after the initial response, so that I don't have to store all quote details on my side.

- The system MAY provide an endpoint to retrieve a saved quote given its quoteId. If so, the client must supply the Quote ID and typically authenticate the same way as when requesting it (so the system knows you have rights to that quote). The request would be something like GET /quotes/{quoteId}. In this request, the quoteId path parameter is required. No other body is needed. The system will look up the quote.

- Upon retrieving, the system will return essentially the same information that was originally provided when the quote was created. This includes the origin/destination, commodity info, and the list of shipping options (or a single option if the quote was specific to one carrier) along with the prices, etc. It allows them to get the quote details again without changing it. The system must ensure the quote hasn't expired; if it has, it might return an error or a flag saying it's expired (or potentially could re-rate it, though that's unusual on a retrieve).

- The retrieve quote call MUST be restricted to authorized users – e.g., you shouldn't be able to randomly query quote IDs that aren't yours. Thus, if a user from a different account attempts to retrieve a quoteId not generated by them, the system should return 404 Not Found or 403 Forbidden. The requirement is essentially security: quote data can contain sensitive pricing info, so it's treated like a secure resource.

Retrieve Quote: Response (Optional)

If the Quote retrieval function is used, the response must return the quote details or an error if not found/expired.

- On a successful GET for a quote, the API would respond with a 200 OK and a body containing the quote information, likely identical in schema to the original quote response. This means a quoteId, possibly a copy of the request parameters (origin, dest, etc.), a rateFound flag, and the list of shippingOptions. Each option again has carrier, service, transit, cost, etc. By providing the same structure, it's easy for the client to reuse the parsing logic. Essentially, it's as if you called the quote fresh, except no changes should occur.

- If the Quote ID does not exist or the user is not authorized, the API MUST return an appropriate error. 404 Not Found is typical if the ID is unknown or no longer exists. 403 Forbidden if the ID exists but belongs to a different account. The response body may say "Quote not found or no longer available."

- If the quote exists but expired, one approach is a 410 Gone status, indicating the resource is gone. However, it's more common just to treat it as not found. Alternatively, the API might still return it with a flag rateFound: false or a field indicating expiration. The product spec should clarify this. Ideally, a shipper should not rely on retrieving an expired quote to see the old price – they should request a new quote. That way an expired quote's ID might just return not found, to avoid confusion with active quotes.

- Any metadata about usage can be included. For example, if the system tracks whether this quote has been converted to an order, it might include a field like rateSelected: true if the quote was used to book a shipment already. That could be helpful to prevent double-booking the same quote.

References

Dates & Date/Times

This API Standard will reference ISO 8601 for all dates and date-times

Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 , RFC 2119 , and RFC 8174 when, and only when, they appear in all capitals, as shown here.

By adhering to these requirements, the Rate Quote API facilitates a smooth, automated way to get pricing and reduces the administrative overhead of negotiating and confirming rates. It essentially digitizes the spot quote process within the broader freight transaction lifecycle, from quote to order to tracking to invoice. The accuracy and reliability of this step are crucial, as it sets the financial expectation for the shipment. The NMFTA standard ensures that all parties speak the same "language" in the quote request/response, making it possible to plug in various carrier systems under one uniform interface for shippers and intermediaries.

Rate Quote Visibility Typical Flow