

SeatMap Cache

A More Complete View Of Seat Availability On Every Flight

SeatMap Cache design and functionality represent a better, more efficient approach to maximize the most up-to-the-minute measure of seating opportunities, thus better serving the broadest range of customers on every flight.

By David Hobt

Developing an accurate and up-to-the-minute seat-map cache is a critical factor in making the very best effort to fill as many seats as possible on every flight. The SeatMap Cache idea and prototype have been developed by Sabre Airline Solutions to effectively address issues relating to an airline being flooded with seat-map requests following a shopping call from an online travel agency (OTA). OTAs can send as many as 1,000 seat-map requests at a time, thus flooding the pre-reserved seat system.

Sabre's research team created a seat-map cache that can be called either in-path (during a shopping request) or out-of-path (after a shopping request). The initial prototype, which has now been running for more than a year, was created for a series of five domestic airlines, four international airlines, 12 domestic markets and 12 international markets.

The prototype started with a window beginning 60 days in the future for a range of 60 days. Each night, a new day was added to the cache and any maps for the current day, and maps two days past the current date were removed, in keeping with traditional Sabre support for date change.

Seat maps are stored by trip leg, and reflect price per seat, seat availability by reservations-booking designator (RBD), seat availability and price by frequent-flyer tier, and the seat maps are also intended in the future to reflect seat availability and price by fare-basis code.

Technologies And Concepts

First and foremost, all seat maps are stored at a flight-leg level, not at a flight-segment level, since all flight segments are made up of the underlying legs.

When a request for a seat map for a given flight, origin and destination is made, the cache examines the line of flight for this flight date to determine which leg maps to combine to create the requested segment map.

SeatMap Cache code consists of a service layer, as well as a utilities-and-reporting layer.

The service layer processes all additions, deletions and modifications of the seat maps; completes the leg-to-segment combination on demand; and performs all nightly maintenance against the cache. The utilities-and-reporting layer consists of a large number of commands to achieve various functions such as:

- Refresh a flight/date or an entire day on demand,
- Validate legs and segments across a line of flight,
- Display green-screen of seat maps in the cache,
- Display by equipment type,
- Find and display change of gauge (a change of aircraft while retaining the same flight number).

Cache Features

Seat maps are dynamically created for requested segments based on the underlying legs.

One or more seat maps may be returned by SeatMap Cache depending on whether or not a flight is a change of gauge, and if the request crosses over the change-of-gauge airport.

Once returned, the seat map contains a vast amount of information about each seat, including:

- Valid classes of service for the seat;
- Price for the seat based on RBD;
- Queries for a large array of attributes (seat location, features, seat level, video-screen view, etc.);
- Dynamic seat requests by less-common attributes;
- Availability of seats together, such that families or business associates can filter by flights that have X-number of seats together.

SeatMap Cache supports change of gauge, all equipment types, date-change flights and codeshare parameters.

Update Frequency

The SeatMap Cache prototype does not have access to organic data, and is populated through Sabre Web Services.

Logic within the cache-retrieval process checks the last update timestamp, and if it is determined to be "stale," this map is not returned to the user, and a separate task is initiated to update the seat map and not show obviously out-of-date information.



▲
THE CONCEPT AND DEVELOPMENT OF SEATMAP CACHE IS AIMED TO SOLVE PROBLEMS RELATING TO AN AIRLINE BEING OVERWHELMED WITH SEAT-MAP REQUESTS FOLLOWING A SHOPPING CALL FROM AN ONLINE TRAVEL AGENCY. THESE AGENCIES CAN SEND AS MANY AS 1,000 SEAT-MAP REQUESTS AT A TIME, THUS INUNDATING THE PRE-RESERVED SEAT SYSTEM.

“Reading days” represent a concept from revenue management that has been incorporated into the SeatMap Cache automatic-update logic. A reading day is a day when the system is “read” because, historically, this number of days before departure sees a spike in booking activity.

For example, on Day 13 (meaning 14 days from today), certain fares close down, as they also do at 21 days, seven days, three days and so forth. So every night in SeatMap Cache, besides rolling in a new horizon day and removing past-date flights, all flights are refreshed for the reading days.

The current reading days (before departure) are 0, 1, 2, 3, 4, 5, 6, 13, 21, 30 and 45. This reading-day parameter is configurable.

Processing Seat-Map Requests

A user or an application can request a seat map from the seat-map service layer by specifying airline, flight number, departure date, departure airport and arrival airport.

The service layer queries the data store for the line of flight for this airline, flight number and date. With the line of flight and the request, the leg seat maps are retrieved for the route from the departure airport to the arrival airport.

Then, a new segment seat map is created (or multiple seat maps are created for change of gauge) by combining each leg in order, ensuring that on any leg on which a seat is unavailable, it becomes unavailable for the segment. On each seat map, the equipment is checked and, if different, a new segment seat map is started to complete the trip.

Regarding nightly file maintenance, the horizon date is always rolled in, old maps are removed and all seat maps on reading days are replaced with fresh versions.

To implement a complete shop-by-seats solution, SeatMap Cache is anticipated to be tied into the reservations system after flight selection but before pricing or other subsequent actions, such that flights could be removed that do not satisfy the seating requirements from the list of flights to process, thus returning more valid options to customers.

Cache Commands

Administrative, nightly and informational commands for SeatMap Cache cover its broad range of seat-mapping functions.

For example, the Query Line of Flight feature goes through the cache for each day and displays to the console the line of flight for every airline/flight/date that exists in the cache.

Additional commands and features include:

- Add Day — The feature gets the current last day in the cache, increment by one day, and adds all flight maps and lines of flight to the cache for the next horizon date.
- Get Flight Legs — The command returns each flight leg, along with data about all legs. If the number in the party is included, the response will also reveal how many options exist for that given party size to sit together.
- Get Seat — The command parameter is followed by the airline code, flight number, date, and seat row and letter (for example, 14A). All information about that seat will be displayed.
- Evaluate Cache — The command parameter may be followed optionally by a day to be evaluated. If no date is specified, the entire cache will be evaluated. The command returns informational lines indicating which flights are in the cache, as well as which flights are missing.
- Find Change of Gauge — The command will find all change-of-gauge flights in the cache and display the airline/flight/data of these flights.
- Find Flights by Equipment — The command returns all flights and dates that use the specified equipment, as delineated by the requester.
- Process Reading Days — The command reads the properties file and performs the reading-day updates.
- Get All Equipment — The command parameter, along with the date, results in all equipment types found in the cache being displayed to the user.
- Restore Map — The command restores or refreshes the specified flight in the cache.
- Restore Day — The command restores the entire day of line-of-flight and seat-map objects for the specified date.
- Add Day from Schedule — The command enables manual performance of the update from the schedule.

Built-In Questions

Specific questions are designed to be answered by SeatMap Cache including:

- What is the total of available seats?
- How many aisles/windows/center seats remain?
- How many exit-row seats remain?
- How many premium seats remain?
- How many preferential seats remain?
- How many no-charge seats remain?
- How many bulkhead seats remain?
- How many pay-for seats remain?
- How many seats together for a specified party size remain?

Another level of information involves total available options for a given party size (purely row-based, as splitting a party requires knowledge of traveler types in the party).

Furthermore, details are available on a specific seat, such as seat 12A (whether the seat is an aisle seat, a premium seat or has other pertinent characteristics).

SeatMap Cache represents a step forward in cache technology for seat inventory and is designed to enable airlines to fill more seats with greater efficiency while delivering on the customer promise to better serve and accommodate passengers on every flight. ASCEND

David Hobt is senior principal for Sabre Research. He can be contacted at david.hobt@sabre.com.